REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	30 March 2007	Final Report, Reporting Period: 1 April 2005 to 31 December 2006

4. TITLE AND SUBTITLE

Advanced Polymer Composite Molding Through Intelligent Process Analysis and Control

5. FUNDING NUMBERS
FA9550-05-1-0230

6. AUTHOR(S)

Bob Minaie

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Wichita State University
Mechanical Engineering Department
1845 N Fairmount Street
Wichita, KS 67260

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFOSR
875 North Randolph Street
Suite 325, Room 3112
Alexandria, VA 22203

Dr Charles Lee (NL)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-SR-AR-TR-07-0163

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release;
distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

To prevent dry spot formation in RTM, a control interface and four different adaptive control algorithms were developed and tested with numerical simulations. The interface is capable of controlling the flow pattern of resin as it fills a mold containing a preform. The mold is equipped with multiple inlet gates, a single vent and a spinal sensor system that continuously feeds the interface with the resin flow front location along the spine lines connecting the inlet gates to the vent. Four different adaptive control algorithms targeting on injection flow rate control, injection pressure control, linearly-corrected pressure control, and the combined flow rate and linearly-corrected pressure control were developed.

Placement of the gates-sensors in addition to the possibility of permeability disturbances (race-tracking) could affect the filling pattern and ultimately the controllability of the developed schemes. To overcome this issue, the location of the gates-sensors was optimized based on the minimization of void formation and filling time. Particle Swarm Optimization (PSO) coupled with adaptive control was implemented on the aforementioned problem and results were compared to previous work using Selective Exhaustive Search (SES) and Genetic Algorithm (GA).

Three journal papers and six conference papers have been published.

14. SUBJECT TERMS	15. NUMBER OF PAGES
	74
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	None

SUMMARY

Resin Transfer Molding (RTM) is a polymer composite manufacturing process that produces high-strength and lightweight parts for aerospace applications. During the filling stage of RTM, uncertainties (such as the race-tracking and resin viscosity changes due to the cure kinetics) exist that could adversely affect the filling pattern by making it non-repeatable from one production run to another, resulting in the formation of dry spots.

To prevent dry spot formation during fabrication of composite parts by RTM, a control interface and four different adaptive control algorithms were developed and tested with numerical simulations. The interface is capable of controlling the flow pattern of resin as it fills a mold containing a preform of fiber reinforcement. The mold is equipped with multiple inlet gates, a single vent and a spinal sensor system that continuously feeds the interface with the resin flow front location along the spine lines connecting the inlet gates to the vent. Four different adaptive control algorithms targeting on injection flow rate control, injection pressure control, linearly-corrected pressure control, and the combined flow rate and linearly-corrected pressure control were proposed and incorporated with the control interface. To provide desirable controllability of the filling process and effective utilization of the resin dispensing equipment, the final formulations were optimized by means of numerical simulations of a RTM part containing different permeability distributions. Section I of the report describes this work.

Placement of the gates-sensors in addition to the possibility of permeability disturbances (race-tracking) could affect the filling pattern and ultimately the controllability of the developed schemes. To overcome this issue, the location of the gates-sensors was optimized based on the minimization of void formation and filling time. Particle Swarm Optimization (PSO) coupled with adaptive control was implemented on the aforementioned problem and results were compared to previous work using Selective Exhaustive Search (SES) and Genetic Algorithm (GA). In addition, after further analysis of the optimization algorithm, an extra term was added to the velocity equation of the PSO algorithm to provide more consistent results that depended on the PSO parameters (maximum velocity, inertia weight, and confidence constants). Section II of the report describes this work.

Three journal papers and six conference papers have been published that are listed below.

1. O. Restrepo, K. Hsiao, A. Rodriguez, B. Minaie, "Development of Adaptive Injection Flow Rate and Pressure Control Algorithms for Resin Transfer Molding," Journal of Composites Part A: Applied Science and Manufacturing (in press).
2. A. Rodriguez, O. Restrepo, B. Minaie, "Optimization of Gate-Spine-Sensor Location in Resin Transfer Molding Using Particle Swarm Optimization Algorithm Coupled with Adaptive Control," 52th International SAMPE Symposium, 2007.
3. P. Kashani, A. Rodriguez, B. Minaie, "Optimization of the Location of Gates and Filling Pattern Sensors Using Genetic Algorithm in Resin Transfer Molding" Proceedings of the SAMPE 2006 Fall Technical Conference, 2006.
4. K. Hsiao, R. Little, O. Restrepo, B. Minaie, "A Study of Direct Cure Kinetics Characterization During Liquid Composite Molding," Journal of Composites Part A: Applied Science and Manufacturing, Vol. 37, No. 6, pp. 925-933, 2006.

5. R. Sadeghian, S. Gangireddy, B. Minaie, K. Hsiao, "Manufacturing Carbon Nanofibers Toughened Polyester/Glass Fiber Composites Using Vacuum Assisted Resin Transfer Molding For Enhancing The Mode-I Delamination Resistance," Journal of Composites Part A: Applied Science and Manufacturing, Vol. 37, No. 10, pp. 1787-1795, 2006.
6. A. Rodriguez, B. Minaie, O. Restrepo, K. Hsiao, "Optimization of Spine Sensor Location in RTM," ASME Paper IMECE2005-81798, 2005.
7. R. Sadeghian, B. Minaie, S. Gangireddy, K.T. Hsiao, "Mode-I Delamination Characterization For Carbon Nanofibers Toughened Polyester/Glassfiber Composites," 50th International SAMPE Symposium, 2005.
8. O. Restrepo, A. Rodriguez, K.T. Hsiao, B. Minaie, "Adaptive Flow Control Of RTM Using Spinal Sensor," 50th International SAMPE Symposium, 2005.
9. K.T. Hsiao, R. Little, O. Restrepo, B. Minaie, "Direct Cure Characterization During Vacuum Assisted Resin Transfer Molding," 50th International SAMPE Symposium, 2005.

SECTION I.

ADAPTIVE INJECTION FLOW RATE AND PRESSURE CONTROL FOR RESIN TRANSFER MOLDING

I.1. INTRODUCTION

Resin transfer molding (RTM) is a polymer composite manufacturing technique that consists of injecting resin into a closed mold cavity containing fiber perform and a following resin curing process to bind the fibers into a solid composite (see Figure 1). After the curing process, the solid net-shaped composite part is carefully demolded. Depending on the part geometry, single or multiple inlet gates and vents in a variety of shapes (point, line, etc.) can be used. As the resin fills the mold, it travels from all inlet gates to designated locations where the flow fronts eventually converge. To enable air inside the cavity to escape, the flow front convergence point or last point to fill (LPF) has to coincide with a vent; otherwise, dry spots may form if the injected resin reaches the vent and blocks the vent before air being completely evacuated from the cavity.

Other liquid composite molding (LCM) variations of RTM apply a vacuum through the vent to evacuate gas and create a pressure gradient that drives resin into the mold. Good examples of resin infusion processes of this type are vacuum assisted resin transfer molding (VARTM) and Seemann Composite Resin Infusion Molding Process (SCRIMP™). All variations of RTM are suitable for the production of structural parts. Nevertheless, RTM uses relatively high injection pressure that makes it cost-effective for medium-volume production; whereas SCRIMP™ and VARTM can be implemented using relatively inexpensive molds, making them suitable for low production volumes [1-6].

Although the aforementioned LCM techniques are extensively used in the industry, they are known to have less than perfect success rate. This is a result of difficulties to predict the resin filling pattern inside the mold. Many researchers have addressed this problem by creating multidimensional simulations of flow in porous media that predict the flow pattern of resin as it fills a given geometry. Darcy's Law and the relationship between Darcy's velocity ($\vec{v}_d$) and the flow front velocity ($\vec{v}_f$) can be used for modeling the resin flow front motion in the fiber preform:

$$\vec{v}_d = -\frac{\mathbf{K}}{\mu} \cdot \nabla P \tag{1}$$

$$\vec{v}_f = \frac{\vec{v}_d}{\phi} \tag{2}$$

where $\mathbf{K}$ is the permeability tensor, μ is the viscosity of the resin, ∇P is the pressure gradient, and ϕ is the porosity of the preform.

Both finite element method (FEM) and finite difference method (FDM) implementations of Darcy's Law have been used to simulate flow in porous media. The filling pattern predictions obtained by numerical simulations have been used to design the vent and gate locations for molds used for RTM, VARTM and SCRIMP™ [2,3,7-14].

Regardless of the research effort dedicated to multidimensional flow in porous media, unpredictable variations in preform permeability, such as racetracking, make each filling process unique; numerical simulations cannot perfectly predict the filling pattern [13]. Racetracking occurs when a small gap between the fiber preform and the mold wall forms a channel in which resin flows faster than in the other regions of the mold cavity. Since the gap size is affected by many parameters such as preform cutting, stacking and placement, the effect is usually non-repeatable and greatly reduces the reliability of RTM. To compensate for the effect of these disturbances, flow control strategies and appropriate sensing devices have been employed to minimize dry spot formation in composite parts [9-22,42].

This section presents an adaptive resin-injection control approach to minimize dry spot formation during LCM regardless of flow disturbances. Different from many simulation-based flow control strategies [9,10,13,14,21,22], this control method does not require pre-established flow simulations to make flow control decisions. The basic concept of this approach is to monitor the position of resin flow front during RTM process and use adaptive control algorithms to adjust the injection flow rate or pressure at the different inlet gates to obtain a desirable filling pattern. To effectively track the resin flow front movement, several linear sensors with spinal arrangement, i.e., spinal sensors, are placed from inlet gates to the vent [2,19]. As the flow front moves inside the mold cavity, its profile intercepts each one of these linear sensors [15,17]. The location of each intersection can be used to describe the flow front's movement. If all spines are connected to a single vent, the ideal mold filling will require the resin flow front to reach the vent along all spines at the same time, and cause the last point to fill (LPF) location to coincide with the vent [3]. This methodology can reduce the complexity of the task of eliminating dry spots; however, it may require very accurate control of the injection pumps to drive the resin flow front with the desired flow pattern.

Based on the basic spinal adaptive control concept, we have developed new control algorithms that provide effective utilization of resin dispensing equipment to steer the resin flow front in RTM. Four different adaptive control algorithms targeting on injection flow rate control, injection pressure control, linearly-corrected pressure control, and a combination of flow rate and linearly-corrected pressure control have been proposed and incorporated into the adaptive control system. The algorithms were tested in numerically simulated RTM processes. The results were compared to highlight their strengths and weaknesses in terms of dry spot size, filling speed, and the minimum requirement of injection pump output power controllability (or responding speed).

I.2. PREVIOUS WORK

I.2.1. Passive Control:

A simple control methodology regarded as "passive control" for RTM is described in [7,13]. The idea is to place vents in all locations where dry spots could be formed. "Passive control" consists of selecting a combination of gates and vents that will completely fill the mold

regardless of the variations in preform permeability. Probabilistic mapping can be used to calculate the likelihood of dry spot formation for a set of possible variations in preform permeability and racetracking. Although this control methodology seems simple, it requires simulating all possible permeability distribution scenarios with corresponding probabilities which have to be determined by experienced users and pilot experiments. Sometimes, it can be difficult to implement since one may not be able to place the vents at simulated locations in real manufacturing environments. Moreover, additional setup and cleanup for the surface imperfection generated by the vents and gates will be necessary. As a result, one needs to find the balance between dry spot size and the minimal number of gates and vents if such a passive control strategy is used [13].

I.2.2. Active Control:

Another control approach is to use auxiliary gates in different combinations to manipulate the flow front to obtain a desired filling pattern. This methodology is regarded in [13] as active control and can be implemented in many different ways. Common practices include consecutive opening/closing of individual gates (sequential control) and changing the flow rate (or pressure) at the injection locations (adaptive control).

Artificial Intelligence (AI) coupled with numerical simulations has been used in [9,10] to optimize sequential RTM flow control systems. This automated computer-generated design approach includes decision making regarding (1) the locations of all injection gates and vents, (2) the locations of point sensors, and (3) the sensor driven control instructions that actuate the valves at inlet gates and vents. A unique dimensionless time vector representing the resin arrival times at different sensors is used in a complex resin flow pattern recognition algorithm based on limited number of sensors and a pre-simulated filling pattern database. This system also applies corrections to the real time filling process by comparing it with the most similar scenario in a database of previous simulations, thus a self-learning intelligent control system may be possible. However, this technique requires an experienced user to estimate the ranges and locations of potential racetracking sites so the computational cost to generate the initial database may be reduced. To minimize the requirement of experienced user, Lawrence et. al. [14] developed a sensor dependence mapping algorithm to make the fluid flow follow a target pattern by using an array of point sensors and auxiliary injection gates.

Direct adaptive control was tested in [19] in a simulated simple 1-dimensional RTM process. In this work, an algorithm controlling the injection pressure calculates the filling error and uses it as control feedback to make the flow front follow a predefined filling pattern. The filling pattern is defined as the expected time and location of the flow front during filling. At each time step, the control system retrieves the distance between inlet and flow front and compares it with the predefined filling pattern to calculate the error and correct the pressure for the simulated 1-dimensional RTM process. In [20], Minaie and Chen used flow front positions traced along three spinal sensors as adaptive control feedback parameters in a time-implicit 2-D simulation of flow in porous media. In this study, the filling pattern uncertainty resulting from the racetracking effect was evaluated by changing the permeability along different mold edges. Model Reference Adaptive System (MRAS) was used as adaptive control strategy to assure that the last point to fill (LPF) location coincides with the vent. The simulated results showed that

this adaptive control method was effective; however, this MRAS control system required a strong inlet pressure controllability for accurately driving the RTM flow control. Such a requirement of pressure controllability could grow significantly if large composite parts were to be manufactured.

Nielsen and Pitchumani [21] developed a system that uses the 2-D flow front location information recorded with a camera and real-time filling simulations to control a RTM process. The system uses the flow front location information provided by the camera to run multiple numerical simulation cases with predefined inlet flow rates to predict the next time step 2-D flow front location. Then, it compares the resulting predictions with the desired flow pattern and finally selects the set of inlet flow rates that produce the best numerical match. The control criterion consists of minimizing the least square nodal position difference between the desired flow front and the simulated prediction and constitutes the first attempt to use on-demand filling simulations as part of the control strategy.

Besides the inlet pressure and flow rate, the dependence between resin viscosity and temperature can be used to control the resin flow during RTM. Johnson and Pitchumani [22] demonstrated such a control methodology by locally heating the areas where the flow is lagging to reduce the viscosity of the resin and compensate for localized permeability variations. A flow front location sensor generates the feedback information, while a carbon susceptor mat embedded in the center of the preform provides the conductive environment where induction heating takes place. This method is capable of controlling VARTM flow based on their numerical simulations; however, special care is required for preventing overheating the resin and triggering the resin gelation during the mold filling.

I.3. METHODOLOGY AND IMPLEMENTATION

In this work, four different new adaptive control algorithms that are aimed at reducing the dry spot as well as effectively utilizing the resin injection system are developed and incorporated into an adaptive spinal RTM flow control system. The algorithms are then numerically tested by simulating an RTM filling process with multiple inlet gates and a single vent (see Figure 2). The control methodology consists of using the resin flow front position measured along spines (lines) connecting the inlet gates to the vent as feedback information for an adaptive control algorithms that adjusts the flow rate and/or pressure at each inlet gate to generate a desired mold filling pattern. To evaluate the performance of the new adaptive control algorithms, the disturbed RTM cases with different permeability distributions in the preform and the racetracking channels are used in the simulated RTM flow control study.

I.3.1. Control Strategy

The proposed control strategy is based on the idea that for most of the filling processes the resin flow front motion inside a mold can be roughly modeled as a circumference that decreases in diameter and gradually converges toward the vent (see Figure 2). According to this simplification, the unfilled region can be compared with a circle centered at the vent with area reduction rate proportional to the total injection flow rate. If one assumes that resin being injected through each inlet gate has strong influence over the section along the corresponding

spine of the collapsing circumference, one can estimate the desired flow rate ($\dot{Q}_i^{t+\Delta t}$) at inlet "i" for next time step as:

$$\dot{Q}_i^{t+\Delta t} \propto \left(d_i^t\right)^2 \tag{3}$$

where d_i^t is the distance between the flow front and the vent along the i^{th} spine at time t.

I.3.1.1. Flow Rate Control Algorithm

To operate the RTM injection flow rate control system at its maximum capacity at all times, we can modify Equation (3) into:

$$\dot{Q}_i^{t+\Delta t} = \dot{Q}_{\max} \cdot \left(\frac{d_i^t}{d_M^t}\right)^n \tag{4}$$

Since $\dot{Q}_{\max}$ is the maximum flow rate of the injection pump and $d_M^t = MAX\{d_1^t, d_2^t, d_3^t\}$, the injection flow rate of any of the injection pumps at next time step will not exceed $\dot{Q}_{\max}$. In addition, this modification forces at least one of the injection pumps to be operated at $\dot{Q}_{\max}$ thus maximizes the utilization of the resin dispensing equipment. To compensate for possible additional constraints and to optimize the effectiveness of the control strategy to correct deviated filling patterns, the quadratic distance term in Equation (3) is substituted by the term $\left(d_i^t / d_M^t\right)$ with an arbitrary exponential power "n" ($n \geq 1$; $n \in$ integer). The optimal selection of exponential power 'n' can be determined by simulating the RTM mold filling process of a given part geometry.

I.3.1.2. Pressure Control Algorithm

While trying to find effective ways to control the flow rate-driven RTM processes, one may consider the pressure-driven RTM processes as well. Such a pressure-driven mold filling control will be useful for some resin infusion processes like VARTM and SCRIMP™. Since this alternative makes the filling process even more prone to be disturbed by permeability variations, developing a pressure control strategy is important. Here, we substitute flow rate ($\dot{Q}_i$) by pressure difference ($P_i - P_o$) in Equation (4) to obtain an alternative control algorithm using pressure as the control parameter.

$$P_i^{t+\Delta t} = \left(P_{\max} - P_o\right) \cdot \left(\frac{d_i^t}{d_M^t}\right)^n + P_o \tag{5}$$

where $P_i^{t+\Delta t}$ is the pressure at the i^{th} gate for the next time step, the subscript M refers to the current maximum value, and $P_{\max}$ and P_o are the maximum allowable pressure and the pressure at the vent, respectively.

I.3.1.3. Linearly-Corrected Pressure Control Algorithm

Using Darcy's Law we can relate flow rate to pressure gradient and derive a better pressure control algorithm from Equation (4). If we assume that to have a pressure control strategy is as effective as its predecessor (i.e., the flow rate control algorithm), the ability of both algorithms to adjust the flow front velocity in a given time step should be equal. From a typical flow rate-driven filling simulation (see Figure 3), one can identify three typical filling stages: (1) as resin initially enters the mold through any given inlet, the flow front advances in the shape of a partial ellipse that grows as resin fills the mold; (2) as resin reaches the racetracking channels at the walls, the spinal flow front velocity remains almost constant; and (3) when the flow front starts converging at the last point to fill (LPF), the flow front becomes a circumference that is centered at the LPF and rapidly decreases in size.

The second stage accounts for the largest interval of the filling process. Darcy's Law for one-dimensional flow in a porous channel can be used to find a relationship between flow rate and pressure assuming that the flow front speed is linearly proportional to the injection flow rate of the corresponding gate. Equation (1) and Equation (2) can be used to relate pressure gradient with the flow front speed during the linear stage and obtain:

$$\frac{(P_i - P_o)}{x_i} \propto \dot{Q}_i \tag{6}$$

in which x_i is the distance between the inlet and the flow front.

By combining Equation (6) and Equation (4), one obtains:

$$P_i^{t+\Delta t} = \left(P_{\max} - P_o\right) \cdot \frac{x_i^t \cdot (d_i^t)^n}{(x_i^t \cdot (d_i^t)^n)_M} + P_o \tag{7}$$

It is important to note that a control algorithm using only Equation (7) can yield high inlet pressures that may cause resin to exit the mold through other inlet gates with lower injection pressures. A simple mechanism to prevent such backflow from occurring is to close the gates immediately where backflow occurs until the next time step.

I.3.1.4. Combined Flow Rate and Linearly-Corrected Pressure Control Algorithm

To further optimize the utilization of the resin dispensing equipment, an algorithm that combines Equations (4) and Equation (7) is proposed as well. The algorithm shifts between flow rate and pressure control to keep at least one of them (either injection flow rate or injection pressure) at the maximum capacity without exceeding predefined pressure and flow rate limits. This algorithm is especially useful in preventing fiber washout. Fiber washout is an undesired

phenomenon in which the position of the fiber preform inside the mold is shifted by the resin flow. It typically occurs as high-speed resin enters the mold; the surface tension and viscous drag force prevent resin from penetrating the preform and causing the preform to move in the direction of the flow. As a result, the area near an inlet can end up being occupied by resin with no fiber reinforcement. Fiber washout can be prevented by carefully controlling the injection flow rates and injection pressures when the resin enters the mold cavity. The flow chart of the combined flow rate and linearly-corrected pressure control algorithm is schematically shown in Figure 4.

I.3.2. Optimization of the Exponential Factor in Different Algorithms

The exponential factor '*n*' in the proposed control algorithms can be optimized by case study using a RTM simulation program. To understand the influence of the exponential factor '*n*' on the control performance, a total of twelve cases accounting for '*n*' from 1 to 12 of each control algorithm are tested in five RTM mold filling simulations consisting of different permeability distribution of the same mold geometry. The optimal exponential factor '*n*' is chosen based on the best overall mold filling performance.

The criteria used to select the best exponentiation factor in each control algorithm includes the following elements: (1) Unfilled mold percentage 'U' at the instant the first spinal flow front reaches the vent perimeter; (2) Equivalent mold filling time 'τ' defined as the time that takes for the first spinal flow front to reach the vent perimeter divided by the corresponding filled mold fraction; and (3) Change rate of pump output power (pump responding speed) $\dot{\Pi} = d(P\dot{Q})/dt$ used to evaluate the cost of using a specific pump to achieve satisfactory mold filling control. Finally, each criterion is assigned a weighting factor 'F' accounting for its overall significance.

$$\mathrm{H}^{k}_{m(n)} = \frac{U_{\max} - U^{k}_{m(n)}}{U_{\max} - U_{opt}} \cdot F_{U} + \frac{\tau_{\max} - \tau^{k}_{m(n)}}{\tau_{\max} - \tau_{opt}} \cdot F_{\tau} + \frac{\dot{\Pi}_{\max} - \dot{\Pi}^{k}_{m(n)}}{\dot{\Pi}_{\max} - \dot{\Pi}_{opt}} \cdot F_{\dot{\Pi}} \qquad (8)$$

Equation (8) represents the objective function (H) used to optimize the value of the exponentiation factor '*n*' for each control algorithm '*m*' in the permeability distribution case '*k*'. The constants $U_{\max}$, $\tau_{\max}$ and $\dot{\Pi}_{\max}$ stand for maximum allowable values for each criterion; U_{opt}, τ_{opt} and $\dot{\Pi}_{opt}$ correspond to the desired optimal values, and F_{U}, F_{τ} and $F_{\dot{\Pi}}$ are their respective weighting factors. Finally, $U^{k}_{m(n)}$, $\tau^{k}_{m(n)}$ and $\dot{\Pi}^{k}_{m(n)}$ are the simulation results for each category obtained using a control algorithm '*m*' with an exponentiation factor '*n*', in the permeability case '*k*'. Subsequently, the control strategy optimization performance average ($\Xi_{m(n)}$) of the control algorithm 'm' with exponentiation factor '*n*' for all permeability distributions $(0 \le k \le l)$ can be defined as:

$$\Xi_{m(n)} = \frac{\sum_{k=0}^{l} \mathrm{H}^{k}_{m(n)}}{l+1} \qquad (9)$$

Note that the operation will be repeated for every combination of exponentiation factor ($1 \le n \le 12$) and control algorithm ($1 \le m \le 4$) in our numerical study. The exponentiation factor 'n' yielding the highest average '$\Xi_{m(n)}$' for a given control algorithm 'm' constitutes the solution of the control algorithm optimization.

I.3.3. Control Interface and 2D RTM Simulation Program:

LabVIEW [23] is used to create an interface that communicates between the control algorithm and the RTM composite manufacturing process. The interface forwards the feedback from spinal sensors in the RTM mold to the algorithm and transmits the control commands, which are decided by the control algorithm, to RTM injection pumps. In the following case study, instead of real RTM processes, the numerical RTM simulations are used. Hence, this control interface is connected to an implicit 2-dimensional Control Volume Finite Element Method (CVFEM) RTM simulation program [20] that is based on Darcy's Law and the following assumptions: (1) density and viscosity are constant; (2) inertial effects and surface tension are negligible; (3) the preform is rigid and never deflects; and (4) as the resin flows through the cavity, it fully saturates the fiber preform. The listed assumptions induce some simulation limitations on the RTM simulations. In RTM, the velocity of resin flow changes dramatically near a point inlet; since Darcy's law and the CVFEM simulation used in this work neglect inertial effects, the flow prediction may not be accurate near the inlets. However, the error introduced is insignificant if one compares the inlet areas with the entire area to be filled in RTM. Similarly, since inertial effects are neglected, the CVFEM simulation used in this work cannot accurately predict the resin behavior near the vent. For this reason, the simulated filling information obtained after the first spine line reaches the vent perimeter is not meaningful and should be discarded.

I.4. CASE STUDY

I.4.1. SIMPLE GEOMETRY CASE STUDY

The objective of this case study is to demonstrate how the proposed control strategy can be optimized for a given simple geometry.

I.4.1.1. Mesh Geometry and Racetracking Cases

Figure 5 shows a diagram illustrating the geometry and permeability distribution regions of the first case study. The selected geometry was defined as a two-dimensional mesh consisting of 2925 nodes and 5624 elements divided into seven different permeability regions for a total volume of $365 \times 185 \times 3 mm^3$. The baseline permeability was set to 2.0×10^{-11} m^2. To simulate racetracking effects, the edge elements were reduced to half of the regular element width and the four areas constituted by them assigned considerably higher permeability values than other regions.

To ensure that the flow rate in the finite element racetracking channel is equal to the flow rate in a simulated racetracking channel with different width, an equivalent permeability of the finite element racetracking channel is proposed:

$$K_\beta \cdot \beta = K_B \cdot B \tag{10}$$

where K_β is the permeability of the numerical model racetracking channel and β is its width. The effective permeability (K_B) of a real racetracking channel can be evaluated from the channel width (B) and the bulk preform permeability (K_{xx}) as [9]:

$$K_B = \frac{1}{2} \cdot K_{xx} + \frac{1}{12} B^2 \tag{11}$$

After using Equation (11) to estimate the effective permeability (K_B) of a sample racetracking channel of width "B", we can use Equation (10) to calculate the equivalent permeability "K_β" to simulate a comparable filling process using a numerical model with a racetracking channel of fixed width "β", which in our model is limited by the element size used. Table 1 shows the equivalent permeabilities (K_β) needed to simulate racetracking channels with different physical widths (B). Table 2 lists the permeability distribution cases used in this numerical case study; the permeability notation is described in Figure 5.

To select the location of the vent, we chose a configuration of three inlet gates (nodes 20, 162 and 102) generating a flow front converging in a single last point to fill (LPF) near the geometrical center of the mold when the nominal permeability distribution case (case 0 in Table 2) was applied. The resin viscosity and the preform porosity were set to 0.8Pa-s and 0.55, respectively. We further assumed that the resin injection pumps were driven by compressed air and had the maximum injection pressure P_{MAX}=600kPa. The simulation was run using the same pressure (600kPa) on all inlets, and the resulting LPF was chosen as the vent location (node 1437). Figure 6 shows the mold configuration and the flow front contours for the nominal case with constant pressure injection (case 0). The maximum flow rate was not constrained for the pure pressure control cases in our study as the mold is small. However, the injection flow rate of the pump must be constrained by its maximum injection pressure. Hence, for the constant flow rate and the flow rate control injection processes, $\dot{Q}_{MAX}$ was set to be $2.74\times10^{-8} m^3/s$ based on the nominal permeability distribution case. On the other hand, for the combined flow rate and linearly-corrected pressure control algorithm, which has the mechanism to prevent pressure from overshooting, a different $\dot{Q}_{MAX}$ value ($=5\times10^{-8} m^3/s$) was assigned for fast injection and fiber washout prevention.

I.4.1.2. Optimization of the Exponentiation Factor

Table 3 lists the values assigned to the constants used in the objective function defined by Equation (8). The optimal value for the unfilled region U_{opt} is equivalent to the unfilled region inside the mold (assuming a perfect filling case) as the flow front reaches the perimeter of the

area underneath the vent. The maximum value U_{max} is assumed as 2% for general applications; although certain applications such as aerospace parts may require less than 1% void. The optimal filling time τ_{OPT} was set as 1000 seconds based on the fact that the nominal case (case 0) with constant injection pressure has the filling time of 1094.3 seconds. Similarly, the maximum filling time τ_{MAX}=2500 seconds was assigned because the nominal case with constant injection flow rate has the filling time of 2179.1 seconds. For the change rate of pump output power, we found it has a value of $5.13\times10^{-4}W/s$ for the nominal permeability case with constant injection flow rate. Hence, we set the optimal values for the change rate of pump output power ($\dot{\Pi}_{opt}$) to be $5\times10^{-4}W/s$. To estimate maximum allowable change rate of pump output power, we assumed that the pump can change its output power by 17% of the average pump power every second. The desired average pump power based on the nominal permeability case with constant injection pressure was found to be $2.03\times10^{-2}W/s$. Thus, we assigned $\dot{\Pi}_{max}=3.5\times10^{-3}W/s$.

I.4.2. COMPLEX GEOMETRY CASE STUDY

To demonstrate that this control method does not require pre-established flow simulations to make effective flow control decisions, the control methodology was used to simulate filling of a mold configuration of a complex geometry with gates and vent locations selected using the judgment of an experienced user. Optimizing the location of gates and vent would result in better controllability of the process. However, the approach used here intends to demonstrate that this control method can be used successfully without pre-established flow simulations.

Figure 7 shows the mold geometry, gates, vent and sensor layout. The geometry was defined as a two-dimensional mesh consisting of 14903 nodes and 28926 elements divided into 17 different permeability regions for a total volume of 3431.9 cm^3. The mesh and permeability distribution definitions for this case were chosen in the same manner as in the first case study. Two permeability distributions were studied: the first configuration corresponds to the nominal case and the second configuration includes racetracking and bulk permeability variations. Figure 8 illustrates the definition of the permeability areas and Table 4 lists the values assigned to each region for each of the two permeability cases considered in this case study.

The gates (nodes 529, 11043 and 14752) and vent (node 5365) configuration was selected arbitrarily using as guidelines the geometrical features of the mold and an educated guess of an experienced user regarding the location of the last point to fill if the mold was filled using constant flow rate. The resin viscosity and the preform porosity were set to 0.1 and 0.5 Pa-s, respectively, and the control exponential factor, n, was set to 9. As in the previous case study, filling pressure was limited to P_{MAX}=600 kPa and the flow rate was the maximum attainable without exceeding the maximum mold filling pressure. For the combined flow rate and linearly-corrected pressure control algorithm, which has the mechanism to prevent the pressure from overshooting, the injection flow rate was constrained to $\dot{Q}_{MAX}=1.03\times10^{-6}m^3/s$ that is the average injection flow rate obtained in a pressure driven simulation.

I.5. RESULTS AND DISCUSSION

I.5.1. Simple Geometry Case Study

The simple geometry case study results are presented as two parts. First, the effect of the exponentiation factor '*n*' in each adaptive RTM control algorithm is studied and the optimal '*n*' value for each control algorithm is selected. Then, the performance of the optimized adaptive RTM control algorithms are compared.

I.5.1.1. Control Algorithm Optimization

The simulated RTM cases listed in Table 2 were controlled by each control algorithm with varying exponentiation factor '*n*'. The best exponentiation factor '*n*' for each control algorithm were determined using Equation (8) and Equation (9). To assess the effectiveness of each control algorithm, the result obtained without any adaptive control will be used as the reference case

.

I.5.1.1.1 Inlet Flow Rate Control Algorithm:

Figure 9 (a) shows the effectiveness of the proposed inlet flow rate control algorithm described by Equation (4) in reducing the unfilled region for different permeability scenarios. For highly asymmetrical permeability distributions, increasing the exponentiation factor '*n*' improved the performance. For less disturbed and more symmetric permeability distributions, large exponentiation factors '*n*' either enhanced or did not considerably harm the performance of the filling process even in comparison with a perfect nominal filling scenario (i.e. no control in the nominal case [10:10:10:10] ? (1:1:1)). Figure 9 (b) shows how the equivalent filling time (t) slightly increases when the exponentiation factor '*n*' becomes larger. Figure 9 (c) presents a comparison regarding the maximum change rate of output power ($\dot{\Pi}$) which gauges the responding speed of the injection pumps required by the control algorithm; it is clear that the required pump responding speed was significantly increased for larger exponentiation factors '*n*'.

Table 5 shows the exponentiation factor '*n*' optimization results for the inlet flow rate control algorithm (see Equation (4)). Based on the objective function described by Equation (8) and Equation (9) and the constants given in Table 3, we find that the best performance was obtained using an exponentiation factor n=9.

I.5.1.1.2 Inlet Pressure Control Algorithm and the Linearly-Corrected Inlet Pressure Control Algorithm:

This section presents the optimization results of the inlet pressure control algorithm described by Equation (5) and the linearly-corrected inlet pressure control algorithm described by Equation (7). Figure 10 shows the effectiveness of the proposed inlet pressure control algorithm to correct disturbances in the flow front pattern. Similarly, Figure 11 shows the effectiveness of the proposed linearly-corrected inlet pressure control algorithm. In Figure 10 (a) and Figure 11 (a), we first noticed that even small disturbances can drastically generate large dry spots in the constant inlet pressure injection cases. Both inlet pressure control algorithms were effective in correcting disturbances in the flow front pattern. For cases with highly asymmetrical permeability distributions, increasing the exponentiation factor '*n*' improved the mold filling; for

less disturbed and more symmetric permeability distributions cases, large exponentiation factors 'n' either enhanced or did not considerably affected the filling process. On the other hand, Figure 10 (b) and Figure 11 (b) also illustrate how the equivalent filling time (t) increases when the exponentiation factor 'n' becomes larger. In general, we can say that enhancing the capability of the system to finely control the filling process usually results in longer filling time. For equivalent maximum process pressures, the pressure control algorithms (Equation (5) and Equation (7)) usually filled the RTM mold faster than flow rate control algorithm (Equation (4)). However, using the inlet flow rate control algorithm, most of the cases resulted in a smaller final unfilled region. Figure 10 (c) and Figure 11 (c) present a comparison regarding the maximum change rate of output power ($\dot{\Pi}$) which gauges the responding speed of the injection pumps required by the control algorithms. In both pressure control algorithms, we cannot see a clear relationship between the exponentiation factor 'n' and the quality of the pump. However, in comparison with the flow rate control algorithm, the pressure control algorithms increased the system demand regarding the change rate of pump output power in about one order of magnitude.

Table 6 and Table 7 present the results obtained using the inlet pressure control algorithm (see Equation (5)) and the results from the linearly-corrected inlet pressure control algorithm (see Equation (7)). For both pressure control algorithms, the best performance was obtained using an exponentiation factor 'n' equal to nine. The linearly-corrected pressure control algorithm was found to have better performance than the simple inlet pressure algorithm except the nominal permeability case that was initially optimized for the constant pressure injection process.

I.5.1.1.3 Combined Flow Rate and Linearly-Corrected Pressure Control Algorithm

Figure 12 (a) shows the effectiveness of the combined flow rate and linearly-corrected inlet pressure control algorithm to steer the flow front to the vent. Like other control algorithms, for highly asymmetrical permeability distributions cases, increasing the exponentiation factor 'n' improved the filling controllability; and large exponentiation factors 'n' either enhanced or did not affect the filling process in the less disturbed and more symmetric cases. This control algorithm produced the higher average reductions in unfilled mold fraction. Figure 12 (b) illustrates how the equivalent filling time (t) increases when the exponentiation factor (n) becomes larger. For the same maximum process pressure, this combined flow rate and linearly-corrected pressure control algorithm filled the mold faster than the flow rate control algorithm but was slower than the pressure control algorithm and the linearly-corrected pressure control algorithm. Figure 12 (c) presents a comparison regarding the maximum change rate of output power ($\dot{\Pi}$) optimization criterion. Again, there is no clear relationship between the exponentiation factor 'n' and the requirement for pump if one chooses the exponentiation factor $6 \le n \le 11$. However, compared with the flow rate control algorithm, this algorithm will need more responsive pumps to accurately control the mold filling process. Table 8 shows the optimization results for the combined inlet flow rate and linearly-corrected inlet pressure control algorithm. The results indicate an optimal exponentiation factor n=9.

I.5.1.2. Comparison Between Four Control Algorithms

Figure 13 presents a graphic comparison of dry spot formation results obtained using the optimized control algorithms for the five different permeability distribution cases considered in this study. The dry spot areas are recorded at the instance the first spinal flow front reaches the vent perimeter since the 2-D RTM simulation program cannot accurately predict the RTM process after this stage. Note that the dry spot size in a real RTM process could be smaller than the simulated results if one keeps pumping resin into the mold after this time.

Table 9 compares the filling times by using constant inlet pressure, constant inlet flow rate and the four optimized adaptive control algorithms. It shows that the pressure control algorithm and the linearly-corrected pressure control algorithm yield the fastest mold filling process. On the other hand, one finds that the combined flow rate and linearly-corrected pressure control algorithm, which uses flow rate as initial control parameter to prevent fiber washout, allows for fast completion of the filling process as well. Table 10 compares the dry spot sizes obtained using different injection control algorithms; one can find that the combined flow rate and linearly-corrected pressure control algorithm has the best average performance in reducing dry spot size.

Figure 14 shows a visual comparison of the flow front contours obtained using different injection control algorithms. After analyzing the results presented in Table 10 and Figure 14, we find that the proposed adaptive control algorithms can effectively minimize the dry spot size and increase the success rate of RTM manufacturing processes. The latest control algorithm (the combined flow rate and linearly-corrected pressure control algorithm) yields outstanding results. Table 10 shows that even for the most adverse case (i.e., case ([10:10:10:600] ? (1:1:1)) this algorithm can significantly reduce the final void content from 17.60% (for constant pressure) to 1.02%.

With respect to the responding speed of the resin-dispensing equipment required by each of the proposed control algorithms, the pressure-driven filling processes are found to be more exigent in equipment responding speed. On the other hand, the maximum change rate of pump output power is significantly reduced for the flow rate-driven filling processes. Table 11 shows a comparison of the maximum change rate of pump output power required for using constant inlet pressure, constant inlet flow rate and each of the optimized control algorithms (n=9).

I.5.2. Complex Geometry Case Study

The performance of different control algorithms was evaluated by comparing the resulting dry spot area or corresponding unfilled mold percentage ,U, at the instant the first spinal flow front reaches the vent. Figure 15 presents a graphic comparison of dry spot formation and filling time obtained for each combination of control algorithm and permeability distribution.

Since the choice of inlets and vent location was relatively good for flow rate driven filling, the dry spot resulting from filling the nominal case using constant flow rate was within the acceptable range. Nonetheless, the dry spot resulting from filling the mold using constant pressure was extremely large. The case where permeability variations were introduced, the Inlet

Pressure Control Algorithm, Linearly-Corrected Pressure Control Algorithm, and Inlet Flow Rate Control Algorithm gave acceptable void contents within 2% limit. In contrast, both constant inlet flow rate and pressure simulations produced voids greatly exceeding this value. Table 12 contains the data obtained for the complex geometry case for comparison with the results obtained with each control algorithm. The algorithm that corrects the inlet pressure using the spinal distance from inlet to flow front had a negative effect on the controllability of the process. This is a logical outcome given that the sensors are not straight.

Implementation of the adaptive control strategies using spine sensors for this complex geometry illustrates that the uncertainties resulting from racetracking permeability variations can be accounted for.

I.6. CONCLUSIONS

In this wok, four different spinal adaptive RTM flow control algorithms were developed and optimized using three performance evaluation factors: dry spot size, filling time, and required pump responding speed. The control parameters (either flow rate or pressure at injection gates) were proposed to be linearly adjusted according to the "n^{th}" power a linear combination of distances from the flow front to the vent along different spine lines. An effective and efficient flow-rate driven RTM control algorithm was first developed; furthermore, it was extended to the pressure driven RTM control algorithm based on Darcy's law and the observation of three different stages of flow front development. Different adaptive RTM flow control algorithms based on flow-rate control, pressure control, linearly-corrected pressure control, and combined flow-rate and linearly-corrected pressure control were presented and numerically studied by using RTM flow simulations. Significant enhancement in the reliability of the RTM process was achieved using the spinal adaptive resin injection control approach equipped with different control algorithms. All control algorithms were found more effective when their exponentiation factor (n) was equal to nine.

In the simple geometry case study, in terms of dry spot size, the combined flow-rate and linearly-corrected pressure control algorithm had the best average performance. As for the filling time, the two pressure-driven control algorithms were faster than the other control algorithms. On the other hand, the flow rate control algorithm demanded the least pump responding speed. However, if one also considers the importance of fiber washout prevention during RTM, the combined flow-rate and linearly-corrected pressure control algorithm should have the most balanced performances in terms of mold filling time, dry spot content, requirement in pump responding speed, and prevention of fiber washout. Nevertheless, the adaptive RTM flow control system needs to be optimized for different manufacturing environments. In such a case, the optimization methodology presented in this work permits one to assign different weighing factors and constants for the objective function to reflect a specific manufacturing environment and obtain the optimized adaptive RTM mold filling control system.

For the complex geometry case study, the results showed that the control algorithms (that produced lower void content in the simple geometry case study) were able to reduce the void content to acceptable levels even for mold configurations that were not optimized. The proposed adaptive control algorithms showed that it is possible to control the RTM flow without relying on flow simulations to make flow control decisions during mold filling. However, as indicated

in our case study, a few simulations (for designing the locations of injection gates and vent and for estimating the controllability and pump requirement) are helpful for predicting and optimizing the performance and cost-effectiveness of the RTM flow control system.

SECTION II.

OPTIMIZATION OF GATE-SPINE-SENSOR LOCATION IN RESIN TRANSFER MOLDING USING PARTICLE SWARM OPTIMIZATION WITH ADAPTIVE CONTROL

II.1. INTRODUCTION

Composite materials continue to be the focus of numerous research studies due to their unique properties. Specific advantages include high strength to weight ratios compared to steel and aluminum, and net-shape finished parts. Resin Transfer Molding (RTM) is a manufacturing process widely used to produce fiber reinforced polymer composite materials. It consists of four consecutive stages: (i) placing the fiber mat, (ii) injecting the resin inside the mold, (iii) curing of the resin, and (iv) demolding of the hardened part.

In RTM, final quality of the part can be affected by certain parameters that are difficult to control. During the injection stage, unpredictable permeability variations in the preform can result in undesired filling patterns that can lead to incomplete wetting of all the fibers. When all sections of the flow front do not converge at the vent locations, the resin can obstruct the vent and the initial entrapped air may not be able to escape the mold cavity. When air is not vented from the cavity, dry spots or voids may form due to incomplete wetting of the fibers. Formation of dry spots (voids) leads to localized reduction of the mechanical, electrical and thermal properties of the part. To ensure a complete fill (no dry spot formation), all sections of the resin flow front have to converge at the vent location at the end of the injection stage so that the Last Point to Fill (LPF) coincides with the vent.

Due to human and machine errors in manufacturing, cutting and placing the fiber mat inside the mold, accurate prediction and control of filling process is not a trivial task. These variations can lead to unpredictable permeability areas inside the preform. Of particular interest in this study, is variations in permeability due to racetracking. Racetracking is a high permeability area on the edge of the preform due to a gap formed between the fiber mat and the wall of the mold. In this region, resin flows faster than in the rest of the preform greatly increasing the flow on the mold walls.

In response to the inability to accurately predict RTM's filling stage, researchers have proposed several methods to control the pathway of the resin during injection. These control methods can be classified into two categories: passive control and active control. In passive control, the approach is to place vents in all the locations where dry spots may form (LPFs). Conversely, in active control, the resin flow front is adjusted during filling by changing the flow rate or pressure at the gates to drive all sections of flow front to the desired LPF where the vent is placed. One promising technology to monitor filling in RTM is the spinal-sensor setup. In this setup, lineal continuous sensors are placed between each gate and the vent of the mold. Each sensor is capable of providing in real time the location of the resin flow front along a straight line between a given inlet and the predetermined vent location. Then at each time step, the inlet flow rates/pressures are recalculated using the sensors feedback and an adaptive control algorithm.

The location of gates, flow sensors and vents greatly affects the filling pattern and therefore the controllability of the injection process. Certain location of the gates, sensors and vents may not be able to accurately describe and/or control the resin filling pattern in the injection stage, thus good or "optimal" location of gates and sensors need to be found in order to increase the controllability of the control scheme. The objective of this work is to implement and test two versions of Particle Swarm Optimization (PSO) algorithm to determine the "optimized" location of a set of three gates and their associated sensors to increase the success rate of the injection process. In this case, the vent was fixed and only the location of gates-sensors was considered. The optimized location of gates-sensors is based on the percentage of void formation and the time required to fill the mold.

To implement the optimization algorithm, PSO was coupled with a 2 dimensional RTM numerical simulation and adaptive control that employs the spinal-sensor configuration to monitor the fill front position. Results are then compared to Selective Exhaustive Search (SES) and Genetic Algorithm (GA) to test the effectiveness of PSO.

II.2. PREVIOUS WORK

II.2.1. Previous Work on Sensing

Several sensing techniques have been used in the past to monitor the fill front in RTM. Among these techniques are optics [24-26], ultrasound [27-28], fluorescence [29], calorimetry [30], and DC resistance [31-34]. All these techniques have to be either embedded or in partial contact with the part to obtain appropriate measurements. This inclusion of the sensors in the part would result in undesired final properties.

Another sensing technique used in adaptive control is Analog Current Dielectrometry (ACD). ACD is capable of measuring fill front position, curing extent, temperature and viscosity and are commercially available as Frequency Dependent Electromagnetic Sensors (FDEMS) [35-39]. However, these sensors are point sensing and also require partial contact with the resin to obtain the measurements.

It has been shown that dielectric sensors can accurately measure the fill front and degree of cure in an RTM process. Skordos et al. [15] developed a sensor that relies on fringing electric field theory to measure fill front position and cure extent. It's major advantage is that the sensor does not have to be embedded in the part to obtain measurements. Nevertheless, the sensor has limited resolution. Rooney et al. [40] also developed a sensor capable of converting capacitance to voltage and correlating it to the flow front position. In this sensor, a combination of point sensing with continuous sensing was used without the requirement of being embedded inside the part.

To measure flow front advancement, viscosity, temperature and cure extent in a continuous (line sensors) non-intrusive manner, lineal sensors are being developed. These sensors use Fringing Electric Field (FEF) effects to obtain flow front position, temperature and cure extent. Hegg et al. [17] proved that by using this technology, accurate measurements of the fill front position can be obtained. Later work by Rowe et al. [42] used these sensors to measure the degree of cure in a polymer composite part.

II.2.2. Previous Work on Optimization of Location of Gates, Sensors, and Vents in RTM

Several methods have been used to optimize the location of gates, sensors and vents in RTM. In 1993, Young [43] implemented Genetic Algorithm (GA) and concluded that if good values of the objective function are used, a good location of gates can be obtained. In 1999, Mathur et al. [44] also used GA and added to the previous conclusion that an optimal solution can be obtained by only evaluating 1% of all the possible arrangements. The authors also found that GA by itself could not find the best possible solution. Further research on optimization using GA was performed by Lin et al. [45] where GA was compared with gradient based algorithms to assess the strengths and weaknesses of the evolutionary algorithm. In this work, the authors used a continuous field to find the best location of gates and vents. They concluded that: (i) GA is not recommended for RTM gate/vent optimization, (ii) single-node modeling of the location is inaccurate for RTM simulation, and (iii) dynamically adapting the finite element mesh as an optimization tool would result in better solutions.

Luo et al. [46] proposed the use of a systematic model of RTM virtual manufacturing environment coupled with Neural Network and GA to find the best location for gates and vents. The Neural Network was trained by data provided by the filling program to create a simplified RTM simulation program. Then, the simplified model was coupled with GA to find the best local optima. The authors found good solutions using a minimal amount of computations. Nonetheless, in this approach, the results rely on how well the Neural Network is trained.

Jiang et al. [47] used a mesh distance-based approach to find the optimal location of gates and vents. The goal was to minimize the maximum distance between gates and vents and prevent void formation. This new approach revealed that by using this new model the computational effort was considerably reduced with respect to those used in Luo et al. [46]. Complex geometries were also tested but it could only handle a fixed quantity of gates and vents.

Gokce et al. [48] proposed a new approach to solve the optimization problem using the Branch and Bound Search (BBS) algorithm. In this work, the algorithm was coupled with a numerical simulation and MATLAB® to locate principal and auxiliary injection ports that would be used to account for the permeability disturbances inside the mold. After applying the optimization program to three different cases, the results showed that: (i) on the first case, BBS was 98% and 59% more efficient than Exhaustive Search (ES) and GA, respectively, (ii) in the second case, BBS only used 4% of the simulations required by ES and 59% less than GA; and (iii) in the third case, BBS was 99% more efficient than ES and 69% more efficient than GA. In terms of accuracy, BBS always found the same results as ES for the geometries studied.

Later work performed by Gokce and Advani [7] proposed a Combinatorial Search (CS) to find the location of vents in the presence of disturbances in Liquid Composite Molding (LCM) after the gates were imposed in the geometry. The authors of this work incorporated a thorough statistical study (Weibull's density function) of racetracking that incorporated set and probability theory to approach the problem. Data from the manufacturing floor was incorporated to feed the forecasting function and calculate the probability of each possible racetracking scenario. Three mold geometries were used to test the algorithm. For the cases studied, CS found the same

results as ES and needed only 10% of the computational time. However, this algorithm can only work for a given set of vents and did not consider the optimization of the vent locations. In additional work by Gokce et al. [49] Cascade Algorithm (CA) was proposed as an optimization technique. This approach combined Branch and Bound Search and Map-Exhaustive Search in the algorithm. In this work, the authors achieved the optimization of gate and vent locations considering a detailed study of racetracking. An advantage of this method is that it incorporated the number of vents as an optimization parameter, though only one gate location was considered.

In 2004 Hsiao and Advani [9] developed an online flow control system that designed the sensor and flow actuation network. Later work by Devillard et al. [10] automated and validated this work. The system used a multi-tier GA to find all the optimization parameters: in the first tier, the design parameters were comprised of the location of gates and vents, the location of the point sensors, the number of pumps and fluid delivery lines, in the second tier, the algorithm selected the control action for each possible fill scenario.

Rodriguez et al. [50] used Selective Exhaustive Search (SES) to optimize the location of gates-sensors in RTM when using a spinal-sensor configuration. This approach used a fixed location of a vent and found the optimized location of the gates by only incorporating geometrically representative points on the boundary. This reduced the search space and less computational effort was needed. Five different racetracking cases were used in this work. The approach worked well except for the fact that even though the search space was considerably reduced, it still needed a significant number of simulations. Later work by Kashani et al. [51] used GA to find the location of the gates for the same case study. It was found that by using GA, an acceptable solution was found while reducing the computational effort by more than 95%.

II.3. METHODOLOGY AND IMPLEMENTATION

To optimize the location of gates and sensors in RTM, two versions of Particle Swarm Optimization (PSO) algorithm were coded in Fortran and tested based on their capability to achieve optimal solutions with reduced computational effort. To test the fitness or performance of each arrangement of gates, sensors and vents, PSO employed a user defined objective function that, in this case, has to be maximized in order to find the optimal solution. The maximization of the objective function requires that the percentage of void and the time required to fill the mold are minimized.

In order to find the percentage of void formation and the time required to fill the mold, adaptive control is coupled with a 2-D RTM filling simulation to continuously adjust the flow front so that the last point to fill coincides with the vent. To monitor the location of the flow front during the simulated filling, spine-like sensor setup is used [2,19,20]. In spine-like sensor setup, lineal sensors are placed between each gate and the vent to provide adaptive control algorithm the location of the flow front in real time as it intersect straight lines (spine line sensors) that joins three inlets with the vent. At every time step, the adaptive control algorithm compares the distances between the flow front and the vent to update the flow rate at each gate. If successful, the control strategy will make the flow front converge to the previously selected vent to prevent void formation and reduce fill time. In this case, the location of the gates, sensors and vents play an important role in the final success of the filling process (maximization of the objective function). Figure 16 shows a mold with a set of lineal sensors that together comprise the spine-

like sensor setup. Note that since the sensors extend from each gate to the vent, the change in the location of the gate or the vent implies relocation of the sensor.

Figure 17 shows the approach used for the optimization procedure. The procedure can be divided into two main parts: (i) pre-processing and (ii) optimization loop. In the pre-processing part, the mold is modeled and meshed using commercial CAD software. After that, the possible location of the gates and the permeability cases are determined. In the second part of the process, PSO algorithm (iterative procedure) is used to test and update the gate-sensor location arrangement considered until convergence is achieved. Further details of PSO algorithm will be given in the following subsections.

II.3.1. RTM Simulation Program

The numerical RTM simulation program [20] is based on Darcy's Law of flow through porous media. To obtain the final equations, the following assumptions were made:

- The density and viscosity are constant.
- Inertia forces are neglected due to slow resin velocities and strong viscous effects.
- Fibers are fully saturated as the resin flows through the cavity.
- Surface tension is negligible compared to viscous forces.

By using the aforementioned assumptions, the simplified model has some limitations and therefore errors associated to the model. However, the errors induced by these limitations are insignificant compared to the whole resin injection process in the mold area.

II.3.2. RTM Control Method

Different adaptive control methods have been used to control the flow front advancement using the spinal-sensor configuration [2,19,20]. In this work, the idea used to control the flow front advancement is that as the resin flows inside the mold from different locations (different gates), resin flow front movement is comparable to a circle decreasing in size (area) and centered at the vent. By changing the flow rate on each gate, the flow front can be steered in such a way that the flow front uniformly arrives to the vent and no air is trapped. Figure 16 shows the details of the spinal-sensor setup.

This control method can be implemented using the following equation, where the resin flow rate at each gate is calculated at every time step using a preset constant flow rate and the distance between the vent and the intersection of the flow front with each spine sensor.

$$\dot{Q}_i^{t+\Delta t} = \dot{Q}_{ref} \cdot \frac{(d_i^t)^2}{\sum_{j=1}^{3} (d_j^t)^2} \tag{12}$$

where $\dot{Q}$ is the flow rate of the i^{th} gate, $\dot{Q}_{ref}$ is the reference flow rate, t is the current time, Δt is the time step size, and d is the distance between the flow front and the vent for the i^{th}

sensor. Figure 18 illustrates the application of the spinal-sensor for one gate and illustrates the physical meaning of the variables.

II.3.3. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is an algorithm developed by Kennedy and Eberhart [52] and Eberhart and Kennedy [53] in 1995 to find the optimal solution of continuous and discrete problems in a solution space. The philosophy of this algorithm relies on mimicking the behavior of a bird flock in their quest to find regions of higher fitness (more food). The algorithm relies on the actions, experience, and communication of a social community trying to find the best position (success) of all the members. To find the best solution, a system of "particles" that are initially sparse in the solution space, exchange information about their "success" in their current and previous positions to make the decision of which direction to move and at what pace.

Since 1995 several modifications to the PSO algorithm have been proposed to increase the success of PSO and to better understand the concept behind it [54-59]. PSO has been applied to engineering and non-engineering problems proving its ability to find optimal solution with low computational cost [60-65]. In addition, research has also been done to compare PSO with other evolutionary algorithms [66-68] to assess the effectiveness of the technique.

PSO algorithm can be summarized in the following steps:

1. Initialize the position and velocity vector of a set of particles in the solution space using the number of gates considered as the dimensions of the vectors.
2. Calculate the fitness of each particle in the swarm using the objective function (Equations 16 and 17).
3. Update the best position for each particle, the best position of the swarm and the best ever position achieved by any particle based on the fitness values obtained in step 2.
4. Calculate the new velocity vector for each particle (Equations 14 or 15).
5. Update the position vector of each particle using the velocity vector and the pseudo time step.
6. Go to step 2 until convergence is achieved.

In this optimization case, a population or swarm of particles representing a small fraction of the total number of gate-sensor arrangements is defined. The position of each particle is represented by the vector $\vec{x}$ with as many dimensions as the number of gates considered for each arrangement. The velocity of each particle is represented by a vector $\vec{v}$ and it influences the movement of each particle. In a flock of i number of particles, the position of each particle can be updated using the following equation,

$$\vec{x}^{\,i}_{k+1} = \vec{x}^{\,i}_{k} + \vec{v}^{\,i}_{k+1} \Delta t_{pso} \qquad (13)$$

where v^{i}_{k+1} is the velocity of the i^{th} particle at pseudo-time step $k+1$, and Δt_{pso} is the size of the pseudo-time step of the optimization algorithm.

In order to ensure stability of the scheme, is important to set a maximum velocity v_{max}. This parameter limits the movement of the particles at every pseudo-time step which helps the algorithm to evade divergence, especially at the beginning iterations; when the values of the velocity are high, the particles can leave the search space and the algorithm may diverge.

In this study, to calculate the velocity vector at each pseudo-time step, two versions of PSO were considered: (1) PSO that contains three terms on the velocity equation and (2) PSO that contains four terms in the velocity equation. Further details of the implementation of the algorithm will be given on the case study section.

II.3.4. Three Term Velocity PSO (TTPSO)

The first version of PSO used in this work is the standard three term velocity PSO. In this algorithm only three terms are used to adjust the velocity of the each particle at every pseudo-time step of the optimization loop. The equation to calculate the velocity vector is

$$\vec{v}^{\,i}_{k+1} = w\,\vec{v}^{\,i}_{k} + c_1 r_1 \frac{\left(\vec{p}^{\,i}_{k} - \vec{x}^{\,i}_{k}\right)}{\Delta t_{pso}} + c_2 r_2 \frac{\left(\vec{p}^{\,g}_{k} - \vec{x}^{\,i}_{k}\right)}{\Delta t_{pso}} \qquad (14)$$

where w is the inertia weight, p^i_k is the best position particle k ever achieved, p^g_k is the position of the best swarm particle at the current pseudo-time step (k), r_1 and r_2 are randomly generated numbers between 0 and 1, and c_1 and c_2 are constants that represent the confidence of the particle in it self and in the swarm, respectively. Higher values of c_1 with respect to c_2 would give more confidence to the particle's experience, conversely, higher values of c_2 would give more confidence to the swarm than to the particle itself.

The second and third terms in equation 14 represent a particle's own memory and a particle's perception about the current success of other swarm particles. The second term of equation 14 has the capability of memorizing the positions where particle i achieved its highest fitness. This position will always influence the direction and the speed (velocity) of particle i until it achieves a higher fitness position. On the other hand, the third term of equation 14 is where the best particle of the swarm influences the rest of the particles by "telling" them the position of the highest fitness at that pseudo-time step. In summary, the second and third term of the velocity equation is where the memory (second term) and communication (third term) of the particles coexist.

Another interesting characteristic of equation 14 is the use of the inertia weight w. For higher values of w the slower the convergence because the new velocity depends on old velocities, thus the particle would take more time to reduce its velocity and come to a complete stop to achieve convergence; the upside of using high values for w is that the particle can explore more search space increasing the probability of finding higher fitness positions Even though only fixed values of w are considered in this work for each optimization run, adaptive inertia weight could be used to cover more search space with less computational time.

II.3.5. Four Term Velocity PSO (FTPSO)

In order to give more stability in terms of results vs. computational effort, a more robust version of PSO is proposed. Fourie et al. [59] proposed replacing, in the velocity equation, the best position of the swarm at current iteration (k) with the best position ever achieved by any particle in the swarm. In this study, instead of replacing the aforementioned terms, a fourth term containing the information of the best position ever achieved by any particle in the flock (p^g) was added to the velocity equation. The new term adds stability and decreases the convergence time. Equation 15 shows the addition of this new term to the velocity equation.

$$v_{k+1}^{i} = wv_{k}^{i} + c_1 r_1 \frac{\left(p_k^i - x_k^i\right)}{\Delta t_{pso}} + c_2 r_2 \frac{\left(p_k^g - x_k^i\right)}{\Delta t_{pso}} + c_3 r_3 \frac{\left(p^g - x_k^i\right)}{\Delta t_{pso}} \tag{15}$$

If equations 14 and 15 are compared in terms of performance, it can be seen that the influence of this new fourth term is already included on the second term of equation 14 in the memory of only one particle that achieved that best result ever. However, the formal inclusion of the best position (highest fitness) ever achieved by any particle to the velocity equation of all particles, greatly enhances the influence of this position on the entire flock and faster and better convergence can be achieved using this modified equation (Equation 15).

II.3.6. Objective Function

To optimize the location of gate-sensor arrangement, a fitness value or "score" is calculated using the same objective function as defined in [50,51]. Each gate-sensor arrangement is evaluated by simulating a series of filling cases with different permeability distributions. A fitness score for each simulation is obtained using the following equation:

$$\Gamma_j^k = \frac{MAV - PV_j^k}{MAV - POV} * VW + \frac{MAT - T_j^k}{MAT - OT} * TW \tag{16}$$

This function considers percentage of void formation as well as the time needed to fill a part. In the expression, MAV is the Maximum Allowable Void, PV is the Percentage of Void corresponding to the j^{th} arrangement, k is the current permeability case, POV is the Percentage of Optimal Void, MAT is the Maximum Allowable Time, T is the total filling Time, OT is the Optimal Time, VW is the Void Weigth constant, and TW is the Time Weight constant.

To find the best of all acceptable or positive solutions, the average of the score obtained in the different permeability distribution cases is computed. A positive average satisfies the selection criteria based of the parameters selected for equation 16 (MAV, POV, VW, MAT, OT, and TW). This explanation can be algebraically represented by equation 17:

$$\Theta_j = \frac{\sum_{k=1}^{NC} \Gamma_j^k}{NC} \tag{17}$$

where *NC* is the number of permeability cases considered in the current arrangement.

II.4. CASE STUDY

TTPSO and FTPSO as described previously, where used to optimize the location of three gate-sensor-coupling and one vent aiming to improve the controllability of a simple RTM process. The size of the mold was 365x185x5 mm^3 and it was discretized in a 2D mesh that consisted of 2925 nodes and 5624 triangular elements. The configuration selected for this optimization uses sensors that provide the location of the flow front intersecting a straight line between each inlet and a predefined vent. The goal is to find the best combination of three gate-sensors so that the objective function is maximized. In this work, the location of the vent was fixed and only the change in gate-sensor location was considered. This mold was also previously used with Selective Exhaustive Search [50] and simple Genetic Algorithm [51]. The results of TTPSO and FTPSO are compared to SES and GA.

In this work as in [50] and [51], the boundary (wall) of the mold was discretized in nodes that corresponded to the vertices of the triangular elements, then, the most significant nodes in terms of geometry were selected to be used as possible gates. These nodes are called principal nodes and reduce to only 2024 the number of possible combinations of 3 gate-sensors and 1 fixed vent if arrangements consisting of repeated node locations are discarded. Figure 19 shows the details of the mesh and selected principal nodes.

Five different racetracking cases were used to model permeability disturbances. Figure 20, shows the racetracking areas and different permeability cases. Four racetracking channels were modeled by increasing the permeability of the 2.5 mm width areas adjacent to each side of the mold. The nominal permeability distribution case (1) was defined by assigning 10 times the bulk permeability to all racetracking regions. In addition to the nominal permeability distribution case, four cases were defined by assigning 30 times the bulk permeability to only one racetracking region in each case while setting the permeability of the remaining three racetracking regions to 10 times the bulk.

To select the vent location, three gates were arbitrarily selected based on the judgment of an experienced operator. A nominal permeability distribution filling case was simulated at constant inlet pressure and the last point to fill was selected as the vent location. Figure 19 also shows the arbitrarily selected gates used to find the vent location.

The values assigned to the objective function parameters and the numerical simulation are shown in table 13a and 13b. Note that in this study, higher importance was given to void formation (80% weight factor) than to the filling time. For practical application is more important to yield parts with acceptable quality than to minimize filling time.

Another important parameter assigned to PSO is the maximum velocity used in the search algorithm. Failure to limit the velocity might result in divergence, especially at the starting iterations. In this work, two values of the maximum velocity constants were considered. For the first case, $v_{max} = 2$ and for the second case $v_{max} = 4$. Using these values will limit the search algorithm to skip no more than 10 and 20 percent of the gates domain (24 principal nodes) on the

next pseudo-time step, respectively. On the other hand, in this study, the same level of confidence was assigned to each term in the velocity equations ($c_1 = c_2 = c_3 = c$). In addition to using two different starting velocities, two values of constant coefficient c were considered: $c = 2$ and $c = 3$.

II.5. RESULTS AND DISCUSSION

The Selective Exhaustive Search [50] solution to the 2024 possible combinations of 3 gate-sensor location and one fixed vent confirmed that the solution of the optimization function is highly non-linear with many local optima. According to the data presented in [50] and [51], only 2.3% of all possible gates-sensors-vent configurations yielded positive and therefore acceptable evaluations scores using the objective function described in section 2.4.

Three indicators were used to measure the effectiveness of PSO in the case study proposed: (1) average value of the objective function, (2) number of simulations, and (3) percentage of success. The first indicator is the average value of the objective function obtained with equations 16 and 17 after averaging the results of 100 optimization runs. The second indicator is the average number of simulations needed to achieve the "optimal" solution to the optimization problem also over 100 optimization runs. Finally, the third indicator measures the percentage of success of PSO at achieving the best possible gate-sensor-vent configuration result in 100 trials (same trials as the other two indicators). All indicators were obtained for different values of the velocity's inertia weight.

Figure 21 shows the results obtained for the first indicator. The graph was generated with data points obtained by computing the average value of the objective function of 100 optimization results acquired using variations of FTPSO and TTPSO (left axis). The graph also shows improvement in percentage obtained using FTPSO with respect to TTPSO search algorithms and comparable settings (right axis). Since FTPSO has an additional term in the velocity equation, the same value for c could not be used because it would deliver higher velocities than its TTPSO. So, in order to account for that change, c had to be adjusted so that the sum of the mean constant terms (when c is multiplied by r_1, or r_2, or r_3) would be equal for FTPSO and TTPSO. This adjustment ensures that the sum of the mean constant terms in the velocity equation are equal for FTPSO and TTPSO so that the comparison in terms of velocity values is the same. In this figure, the triangular marker shows the average values for runs in which the sum of the mean constant terms was equal to two, on the other hand, the square marker refers to runs made using values of c in which the sum of the mean constant terms in the velocity equation was equal to three.

In Figure 21a it can be seen that when $v_{max} = 4$ the percentage of change between FTPSO and TTPSO was always positive. This means that FTPSO was always able to find better solutions than TTPSO. Furthermore, beyond the 0.65 inertia weight value, the tendency in the change between four and three term velocity PSO increased with higher inertia weight meaning that FTPSO is more effective than TTPSO with increasing values of inertia weight. On the other hand, in Figure 21b ($v_{max} = 2$) it is seen that the value of the average objective function, in general, is lower than when $v_{max} = 4$ (Figure 21a). Also, the percentage of change between

FTPSO and TTPSO is lower for $v_{max} = 2$ than for $v_{max} = 4$. The overall tendency is that FTPSO generates better average objective function results than TTPSO; the only exception is the case where the inertia weight was set to 0.99 and $v_{max} = 2$ (Figure 21b). These results confirm that FTPSO is more stable and more robust, in terms of finding the best solution, than TTPSO.

To assess the efficiency of TTPSO and FTPSO, the average number of simulations needed to achieve the results shown in Figure 21 are plotted in Figure 22. In Figure 22a, for $v_{max} = 4$, it is clear that fewer simulations are needed for the FTPSO than the standard version of PSO. In fact, as the inertia weight increases, the difference in number of simulations used by each version of PSO also increases. Therefore, it can be concluded that FTPSO finds better results (as seen in Figure 21) with less computational effort than TTPSO (up to 25% less computational effort). Regarding the results obtained for $v_{max} = 2$, less computations are needed for FTPSO than for TTPSO to achieve better results (Figure 22b), though the performance difference between the two versions of PSO is smaller for $v_{max} = 2$ than for $v_{max} = 4$ (Figure 22a). Hence, from Figure 22a and 23b, it can be said that FTPSO performs better than TTPSO, especially when $v_{max} = 4$.

The third indicator used to obtain better insight in the performance of PSO is plotted in Figure 23. As stated before, each version of the search algorithm was tested 100 times. Figure 23 illustrates the percentage of times each algorithm found the best possible solution. For $v_{max} = 4$, FTPSO was able to achieve the best possible solution far more times than TTPSO. Furthermore, for $w = 0.99$, FTPSO had 60% of success rate compared to approximately 26% for TTPSO. Note that as the inertia weight increases, so does the difference in success rate between the two PSO algorithms. In fact, the percentage of change (difference between FTPSO and TTPSO) increased from about 100% to over 250% when the values of the inertia weight varied form 0.35 to 0.99, respectively. In the case of $v_{max} = 2$, for higher values of c (square marker) the results are better for FTPSO than TTPSO, conversely, for lower values of c (triangular marker) the values are better for TTPSO than FTPSO for medium values of inertia weight (from 0.65 to 0.85).

In order to compare the performance of TTPSO and FTPSO with SES and GA, Table 14 shows the results obtained using Selective Exhaustive Search [50], simple GA [51] and two different cases of four and three term PSO for the same case study. In terms of the average value of the objective function, in 100 search trials, all PSO versions performed better than GA; better performance of PSO was obtained for higher values of the inertia weight. It is important to mention that the values shown for GA were obtained over 20 trials and not over 100. On the other hand, for PSO using high inertia values required in average greater number of simulations than GA to find a solution. Therefore, to compare the effectiveness of the algorithms when using approximately the same number of simulations, a case where $w = 0.65$ for FTPSO and TTPSO is shown. Results show that FTPSO performed better than GA in all three indicators. In fact, the computational effort was reduced by approximately 9.5% with respect to GA and the percentage of success was increased more than two fold, thus obtaining better values for the objective function. On the other hand, when using TTPSO for the same case ($w = 0.65$), the algorithm found a better solution than GA but with higher computational cost. Note that even for this case the percentage of success was doubled with respect to GA.

II.6. CONCLUSIONS

In this work two versions of Particle Swarm Optimization where implemented in the optimization of the location of gates-sensors in RTM. These two versions consist of the standard three term velocity PSO and a proposed modified four term velocity PSO. Results show that Particle Swarm Optimization algorithm can perform very well in optimizing the location of gates-sensor in an RTM mold. In the case study, it was found that four term PSO is more robust and can find near to optimal solutions with less computational effort than three term PSO for high velocity values ($v_{max} = 4$). However, this observation does not hold for lower values of v_{max}.

The velocity vector in PSO can be regarded as the mobility that a particle has in a certain timeframe. In this work the mobility was restricted by a maximum velocity v_{max} equal to four and two. In this study, it was found that higher mobility values generated better results; however, if the mobility is too high the probability of slower convergence or even divergence increases. Further research has to be done to obtain optimized values of particle mobility.

The results obtained in this work were also compared to SES and GA to assess the effectiveness of the algorithm. For the case studied in this work, PSO performed better than simple GA. It was found that better average values of the objective function can be achieved with lower computational effort (10%). The approach in this work could also be applied to complex geometries. Efforts are being made to extend this work to the optimization of gates-sensors and vent location in complex geometries.

REFERENCES

1. Seemann WH. Plastic transfer molding techniques for the production of fiber reinforced plastic structures. *U.S. Patent* 4,902,215. February 20, 1990.
2. Restrepo O, Rodríguez A, Hsiao K-T, Minaie B. Adaptive Flow Control of RTM Using Spinal Sensors. Proceedings of the 50th SAMPE Symposium and Exhibition. Long Beach, California, May, 2005.
3. Li W. Process Analysis and Adaptive Control of Resin Transfer Molding, Ph.D. dissertation. Birmingham, Alabama: University of Alabama at Birmingham, 2001.
4. Mazumdar SK. Composites Manufacturing. Boca Raton: CRC Press, 2002.
5. Walsh SM, Mohan RV. Sensor-based control of flow fronts in vacuum-assisted RTM. Plastics Engineering 1999; 55(10):29-32.
6. Parnas RS, Walsh SM. Vacuum-assisted resin transfer molding model. Polymer Composites 2005;26(4):477-485.
7. Gokce A, Advani SG. Combinatorial Search to Optimize Vent Locations In The Presence of Disturbances In Liquid Composite Molding Processes. Materials and Manufacturing Processes 2003;18(2):261-285.
8. Hang CC, Lee TH, Ho WK. Adaptive Control. Research Triangle Park, North Carolina: Instruments Society of America, 1993.
9. Hsiao K-T, Advani SG. Flow sensing and control strategies to address race-tracking disturbances in resin transfer molding---Part I: design and algorithm development. Composites Part A: Applied Science and Manufacturing 2004;35(10):1149-1159.
10. Devillard M, Hsiao K-T, Advani SG. Flow sensing and control strategies to address race-tracking disturbances in resin transfer molding---Part II: automation and validation. Composites Part A: Applied Science and Manufacturing 2005;36(11):1581-1589.
11. Panitapu R. Analysis of Filling Pattern in Resin Transfer Molding, Masters Thesis. Mobile, Alabama: University of South Alabama, 2003.
12. Shojaei A. Numerical simulation of three-dimensional flow and analysis of filling process in compression resin transfer moulding. Composites Part A: Applied Science and Manufacturing, In Press, Corrected Proof, Available online 11 November 2005
13. Lawrence J, Hsiao K-T. Don R, Simacek P, Estrada G, Sozer E, Stadtfeld H, Advani SG. An approach to couple mold design and on-line control to manufacture complex composite parts by resin transfer molding. Composites Part A: Applied Science and Manufacturing 2002;33(7):981-990.
14. Lawrence JM, Hughes P, Advani SG. Experimental Validation of Dependence Map based Control in Liquid Composite Molding. Journal of Composite Materials 2006;40(13):1137-1162.
15. Skordos AA, Karkanas PI, Partridge IK. A Dielectric Sensor for Measuring Flow in Resin Transfer Moulding. Measurement Science and Technology 2000;11(1):25-31.0.
16. Kim HG, Lee DG. Dielectric cure monitoring for glass/polymer prepreg composites. Composite Structures 2002;57(1-4):91-99.
17. Hegg MC. Monitoring of Resin Transfer Molding Processes with Distributed Dielectric Sensors, Masters Thesis. Seattle, Washington: University of Washington, 2004.
18. Stöven T, Weyrauch F, Mitschang P, Nitzel M. Continuous monitoring of three-dimentional resin flow through a fiber Preform, Composites Part A: Applied Science and Manufacturing 2003;34(6):475-480.

19. Minaie B, Li W, Jiang S, Hsiao K-T, Little R. Adaptive Control of Non-Isothermal Filling in Resin Transfer Molding. Proceedings of the 49th SAMPE Symposium and Exhibition. Long Beach, California, May 2004
20. Minaie B, Chen YF. Adaptive Control of Filling Pattern in Resin Transfer Molding Process. Journal of Composite Materials 2005:39(16);1497-1513.
21. Nielsen DR, Pitchumani R. Closed-loop control in resin transfer molding using real-time numerical process simulations. Composites Science and Technology 2002:62(2);283-298.
22. Johnson R, Pitchumani R. Simulated Active Flow Control in a VARTM Process Using Localized Induction Heating. ICCM-14 Proceedings. San Diego, California, July 14-18, 2003.
23. National Instruments, National Instruments™ LabVIEW™ Getting Started With LabVIEW, Part Number 321527E-01, November 2001.
24. Ahn SH, Lee WI. Measurement of the 3-Dimensional Permeability of Fiber Preforms Using Embedded Fiber Optic Sensors. Journal of Composite Materials 1995; 29(6): 714-733.
25. Bernstein JR, Wagner JW. Fiber Optic Sensors for Use In Monitoring Flow Front in Vacuum Resin Transfer Molding Processes. Review of Scientific Instruments 1997; 68(5): 2156-2157.
26. Crosby PA, Powell GR, Fernando GF, Waters DN, France CM, Spooncer RC. Comparative Study of Optical Fiber Cure Monitoring Methods. SPIE The International Society for Optical Engineering 1997; 3042: 141-153 (1997).
27. Shepard DD, Smith KR. Ultrasonic Cure Monitoring of Advanced Composites. Sensor Review 1999; 19(3): 187-191.
28. Fomitchov PA, Kim YK, Kromine AK, Krishnaswamy S. Laser Ultrasonic Array System for Real-Time Cure Monitoring of Polymer-Matrix Composites. Journal of Composite Materials 2002; 36(15): 1889-1901.
29. Woerdeman DL, Parnas RS. Cure Monitoring in RTM Using Fluorescence. Plastics Engineering 1995; 51(10): 25.
30. Tavakoli SM, Avella M, Phillips MG. Fiber Resin Compatibility in Glass Phenolic Laminating Systems. Composites Science and Technology 1990; 39(2): 127-145.
31. Barooah P, Berker B, Sun JW. Lineal Sensors For Liquid Injection Molding of Advanced Composite Materials. Journal of Materials Processing and Manufacturing 1998; 6(3): 169-184.
32. Lee CW, Rice BP, Buczek M, Mason D. Resin Transfer Process Monitoring and Control. SAMPE Journal 1998; 34(6): 48-55.
33. Schwab SD, Levy RL, Glover GG. Sensor System for Monitoring Impregnation and Cure During Resin Transfer Molding. Polymer Composites 1996; 17(2): 312-316.
34. Shepard DD. Resin Flow Front and Cure Detection in RTM and Other Resin Infusion Processes. Proceedings of the 30th International SAMPE Technical Conference 1998; 351-360.
35. Kranbuehl DE, Hood D, Kriss A, Barksdale R, Loos AC, Macrae JD, Hasko G. In-Situ FDEMS Sensing and Modeling of Epoxy Infiltration, Viscosity and Degree of Cure During Resin Transfer Molding of A Textile Preform. Polymeric Materials Science Engineering 1996; 74: 28-29.
36. Kranbuehl DE, Hoff M, Eichinger D, Clark R, Loos A. Monitoring And Modeling The Cure Processing Properties Of Resin Transfer Molding Resins. International SAMPE Symposium and Exhibition 1989; 34: 416-425.

37. Kranbuehl DE, Kingsley P, Hart S, Hasko G, Dexter B, Loos AC. In-Situ Sensor Monitoring And Intelligent Control Of The Resin Transfer Molding Process. Polymer Composites 1994; 15(4): 299-305.
38. Kranbuehl DE, Williamson A, Loos AC. Sensor-Model In-Situ Control Of The RTM Composite Process. Proceedings of 36th International SAMPE Symposium and Exhibition 1991; 536-545.
39. Kranbuehl DE, Hoff M, Eichinger D, Clark R, Loos AC. Monitoring and Modeling the Cure Processing Properties of Resin Transfer Molding Resins. Proceedings of the 34th International SAMPE Symposium and Exhibition 1989; 416-425.
40. Rooney M, Biermann PJ, Carkhuff BG, Shires DR, Mohan RV. Development of In-Process RTM Sensors for Thick Composite Sections. Proceedings of 1998 American Control Conference 1998; 6: 3875-3878.
41. Hegg M, Ogale A, Mescher A, Mamishev A, Minaie B. Remote Monitoring of Resin Transfer Molding Processes by Distributed Dielectric Sensors. Journal of Composite Materials 2005; 39(17): 1519-1530.
42. Rowe G, Yi J, Chiu K, Tan J, Mamishev A, Minaie B. Fill-Front and Cure Progress Monitoring for VARTM with Auto-Calibrating Dielectric Sensors. Proceedings of SAMPE 2005 International Symposium and Exhibition 2005; 50: 2793-2806.
43. Young WB. Gate Location Optimization in Liquid Composite Molding Using Genetic Algorithm. Journal of Composite Materials 1994; 28(12): 1098-1113.
44. Mathur R, Fink B, Advani S. Use of Genetic Algorithm to Optimize Gate and Vent Locations for the Resin Transfer Molding Process. Polymer Composites 1999; 20(2): 167-178.
45. Lin MY, Murphy MJ, Hahn HT. Resin Transfer Molding Process Optimization. Composites: Part A 2000; 31(4): 361-371.
46. Luo J, Liang Z, Zhang C, Wang B. Optimum Tooling Design for Resin Transfer Molding with Virtual Manufacturing and Artificial Intelligence. Composites: Part A 2001; 32(6): 877-888.
47. Jiang S, Zhang C, Wang B. Optimum Arrangement of Gate and Vent Locations for RTM Process Design Using a Mesh Distance Approach. Composites: Part A 2002; 33(4): 471-481.
48. Gokce A, Hsiao KT, Advani SG. Branch and Bound Search to Optimize Injection Gate Locations in Liquid Composite Molding Processes. Composites: Part A 2002; 33(9): 1263-1272.
49. Gokce A, Advani SG. Simultaneous Gate and Vent Optimization in Liquid Composite Molding. Composites: Part A 2004; 35(12): 1419-1432.
50. Rodriguez A, Minaie B, Restrepo O, Hsiao KT. Optimization of Spine Sensor Location in Resin Transfer Molding. Proceedings of IMECE: ASME Dynamic Systems and Control Division 2005; 74(1 Part A): 687-695.
51. Kashani P, Rodriguez A, Minaie B. Optimization of the Location of Gates and Filling Pattern Sensors Using Genetic Algorithm in Resin Transfer Molding. SAMPE Technical Conference 1996; in press.
52. J. Kennedy, R. Eberhart. Particle Swarm Optimization. Proceedings of IEEE International Conference on Neural Networks 1995; 44: 1942-1948.
53. Eberhart R, Kennedy J. A New Optimizer Using Particle Swarm Theory. Proceedings of the sixth International Symposium and Micro Machine and Human Science 1995; 39-43.

54. Shi Y, Eberhart R. . Proceedings of the International Conference on Evolutionary Computation 1998; 66-73.
55. Shi Y, Eberhart R. . Evolutionary Programming VII: Proceedings EP98, Lecture Notes in Computer Science 1998; 591-600.
56. Eberhart R, Shi Y. Particle Swarm Optimization: Developments, Applications and Resources. Proceedings of the 2001 Congress on Evolutionary Computation 2001; 1: 81-86.
57. Yang C, Simon D. . IEEE International Conference on Systems Engineering, 18.
58. He S, Prempain E, Wu QH. . Engineering Optimization 2004; 36(5): 585-605.
59. Fourie PC, Groenwold AA. The Particle Swarm Optimization in Size and Shape Optimization. Structural and Multidisciplinary Optimization 2002; 23(4): 259-267.
60. Venter G, Sobieszczanski-Sobieski J. Particle Swarm Optimization. AIAA Journal 2003; 41(8): 1583-1589.
61. Brandstatter B, Baumgartner U. Particle Swarm Optimization - Mass-Spring System Analogon. IEEE Transactions on Magnetics 2002; 38(2): 997-1000.
62. Parsopoulos KE, Vrahatis MN. . Advances in Natural Computation; First International Conference 2005; 582-591.
63. Lin JH. Particle Swarm Optimization For Control Of Nonlinear Dynamics. Proceedings of the First International Conference on Innovative Computing, Information and Control 2006.
64. Chen YC, Wang HC, Su TJ. . International Conference on Innovative Computing, Information and Control 2006.
65. Nenortaite J, Simutis R. . 5th International Conference on Intelligent Systems Design and Applications 2005; 520-525.
66. Elbeltagi E, Hegazy T, Grierson D. . Advanced Engineering Informatics 2005; 19(1): 43-53.
67. Eberhart R, Shi Y. . Evolutionary Programming VII. Proceedings of the 7th International Conference 1998; EP98: 611-616.
68. Angeline P. Evolutionary Optimization Versus Particle Swarm Optimization: Philosophy and Performance Differences. Lecture Notes in Computer Science 1998; 1447: 601.

LIST OF FIGURES

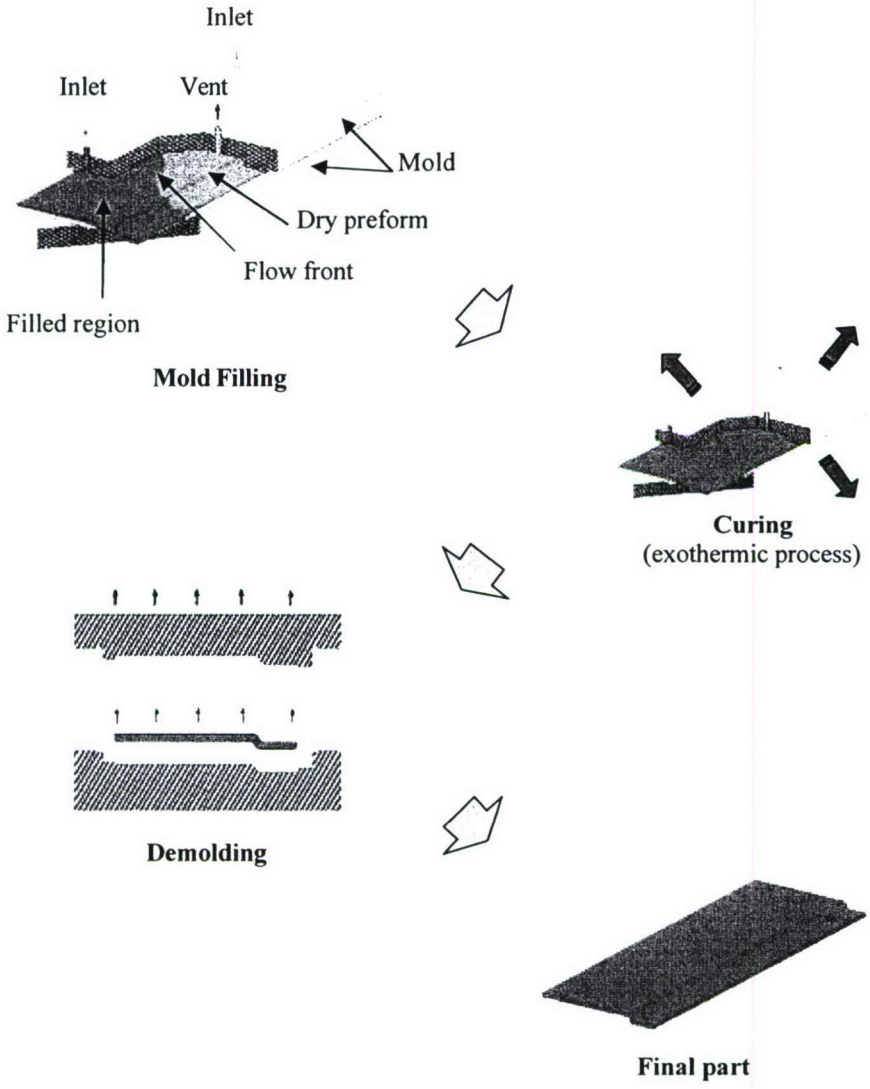


Figure 1. Resin Transfer Molding (RTM).

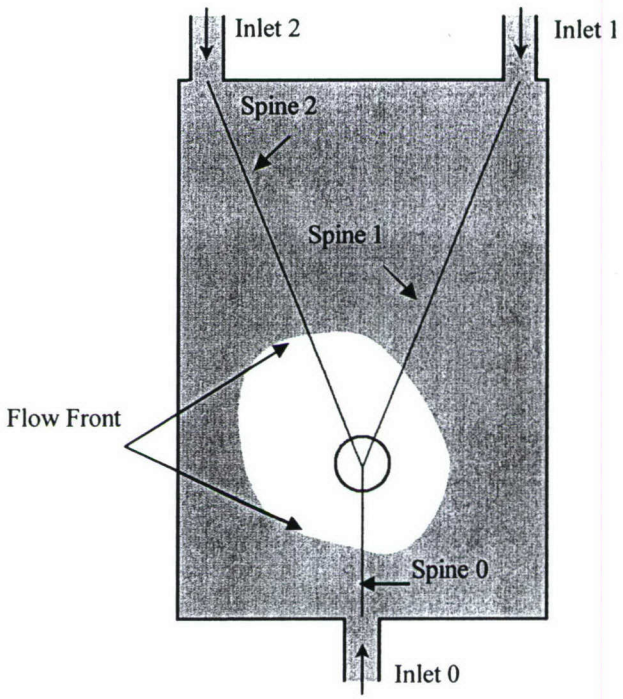
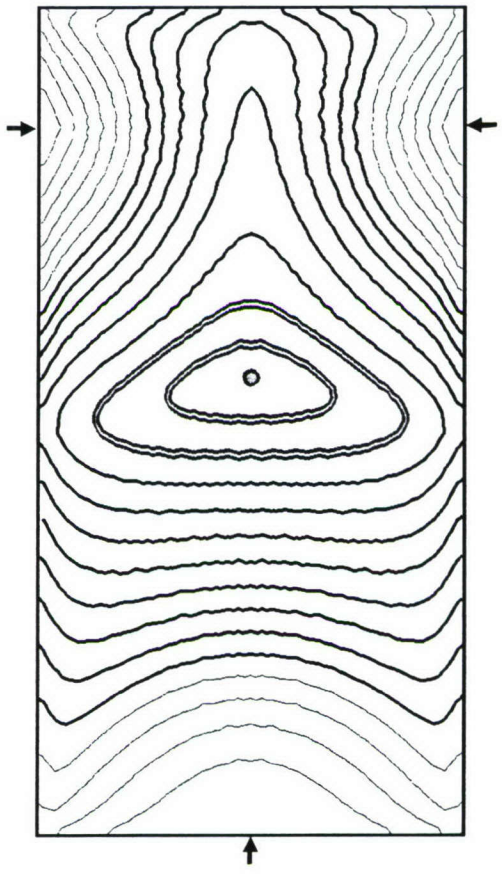


Figure 2. Mold configuration and its elements.



Inlet Gates
Mold wall
Vent
1st stage flow front contours (ellipse)
2nd stage flow front contours (linear)
3rd stage flow front contours (circumference)

Figure 3. Three filling stages in spinal adaptive control.

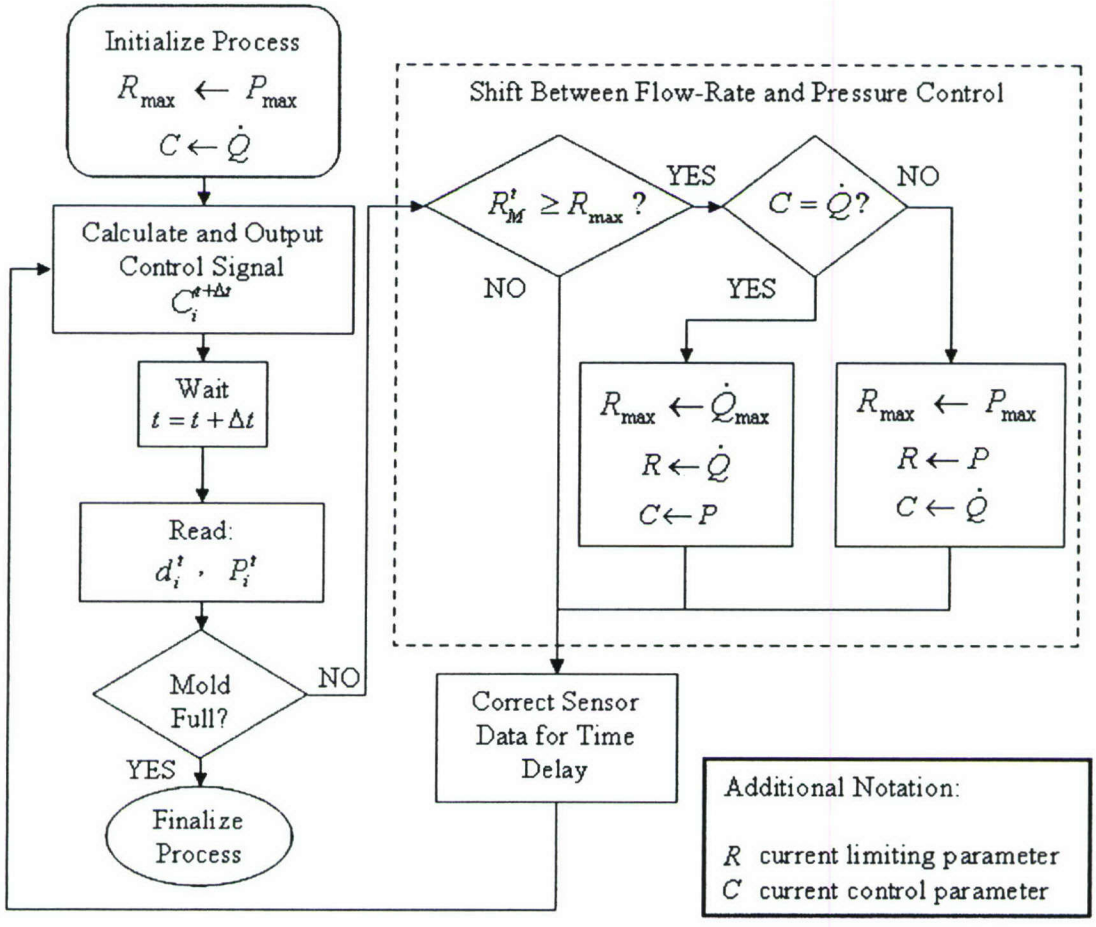


Figure 4. Flow chart of the combined flow rate and linearly-corrected pressure control algorithm.

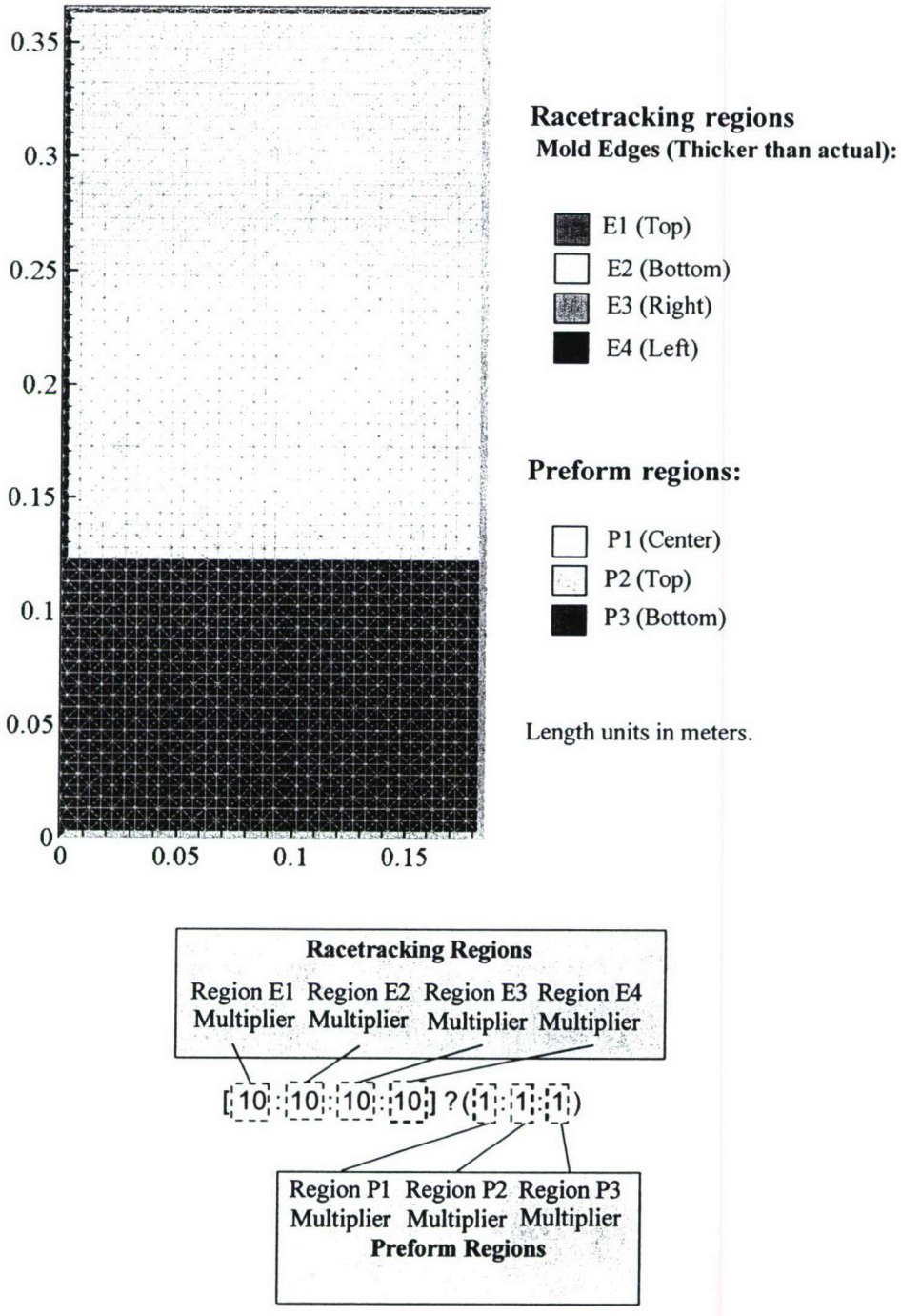


Figure 5. Simple geometry mesh, preform permeability regions, and permeability distribution notation.

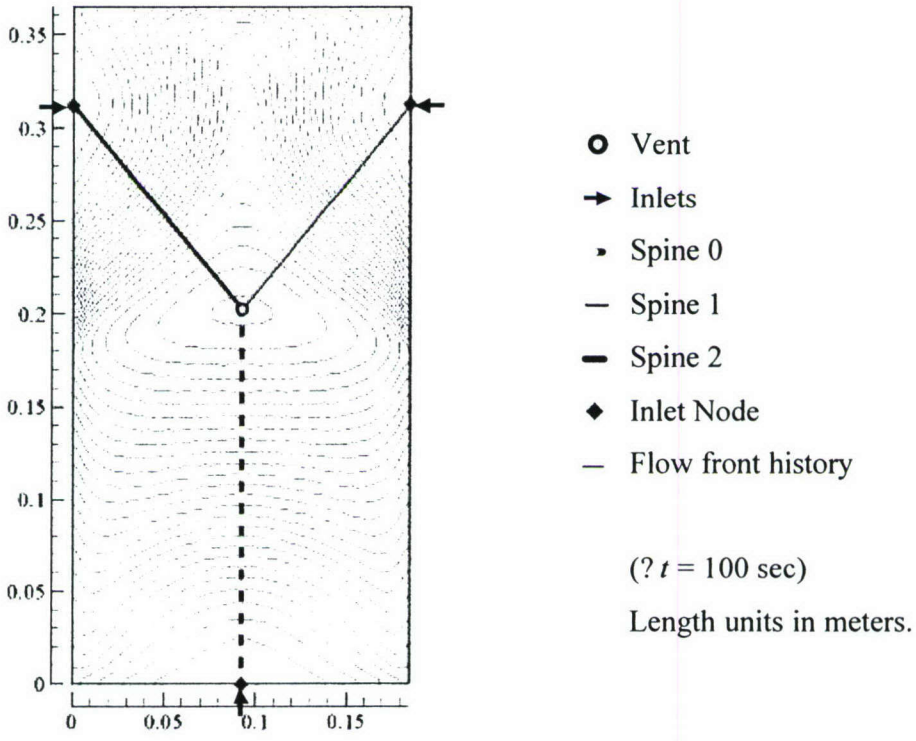
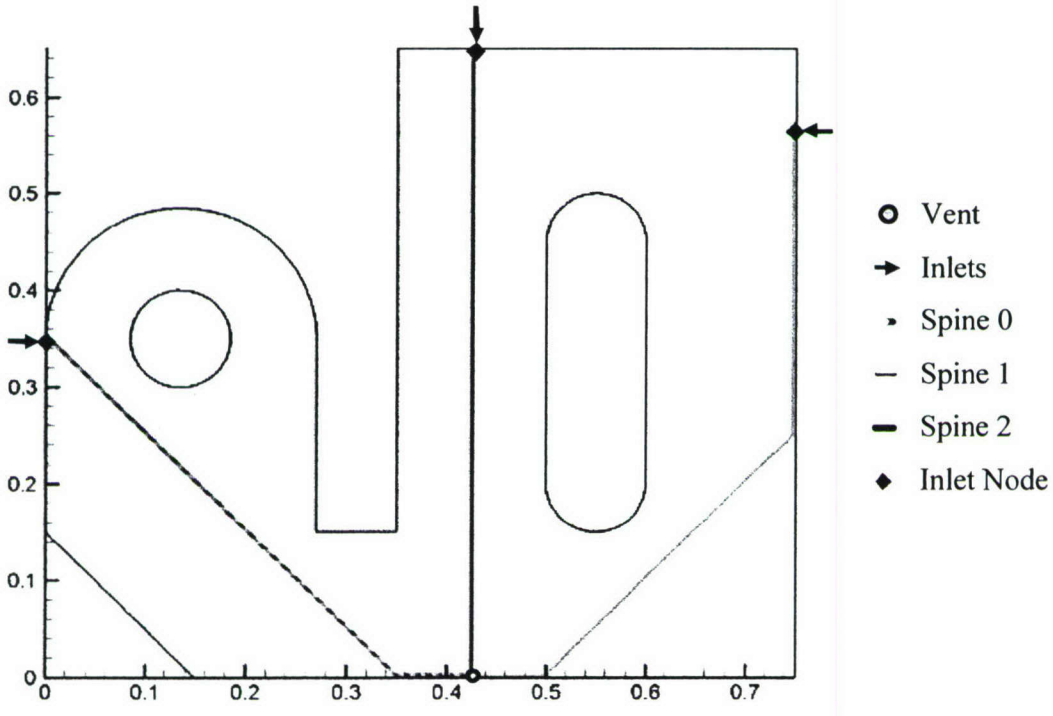


Figure 6. Simple geometry mold configuration and nominal flow front contours (? t = 100 sec).



Length units in meters.

Figure 7. Complex geometry mold configuration.

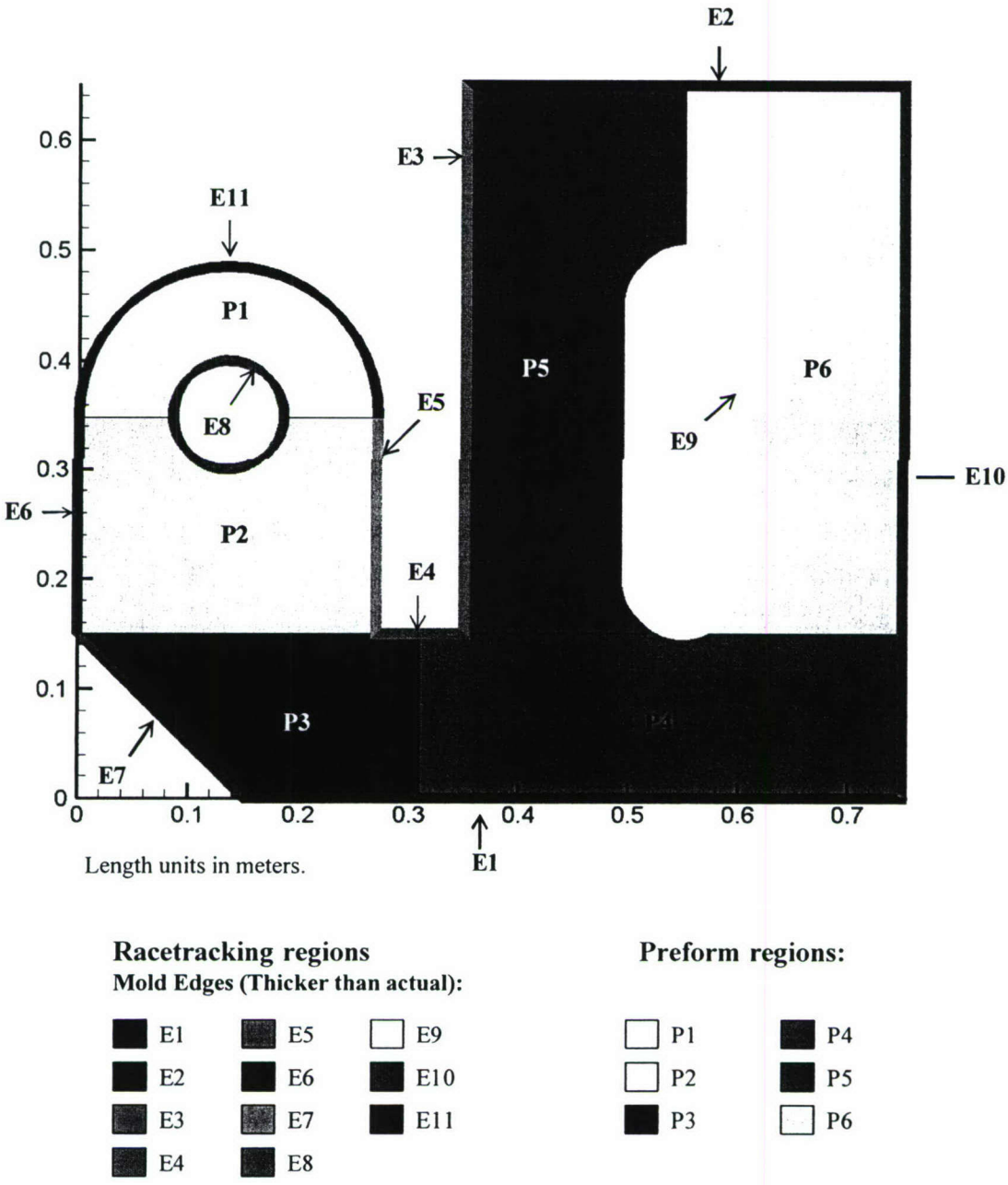


Figure 8. Complex geometry permeability regions.

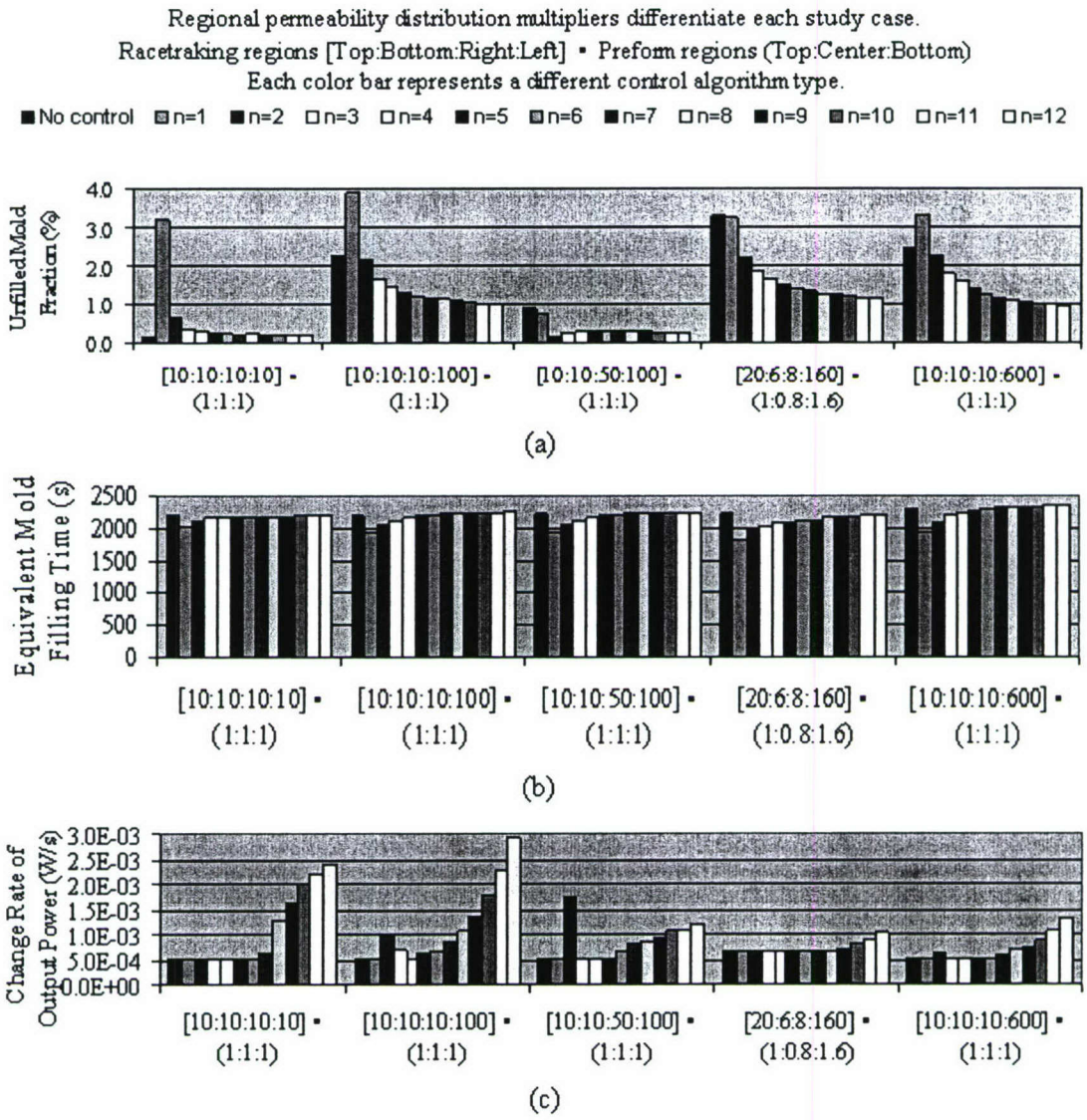


Figure 9. Flow rate-driven RTM control process (Equation (4)): (a) comparison of unfilled mold percentage (U), (b) comparison of equivalent filling time (t), and (c) comparison of maximum change rate of pump output power ($\dot{\Pi}$). Note that the unfilled mold fraction was determined at the instant the first spinal flow front reached the vent perimeter.

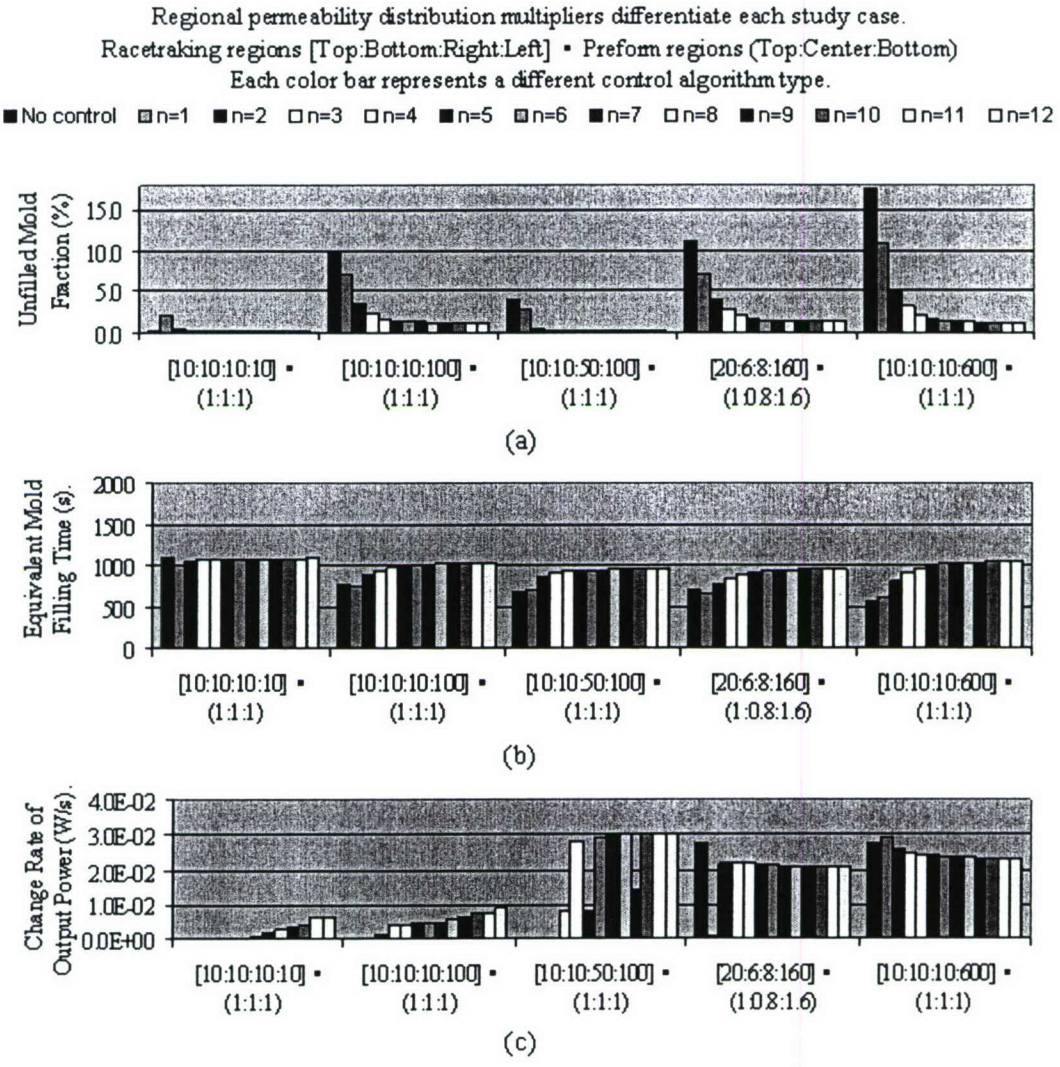


Figure 10. Inlet pressure driven RTM control process (Equation (5)): (a) comparison of unfilled mold percentage (U), (b) comparison of equivalent filling time (t), and (c) comparison of maximum change rate of pump output power ($\dot{\Pi}$). Note that the unfilled mold fraction was determined at the instant the first spinal flow front reached the vent perimeter.

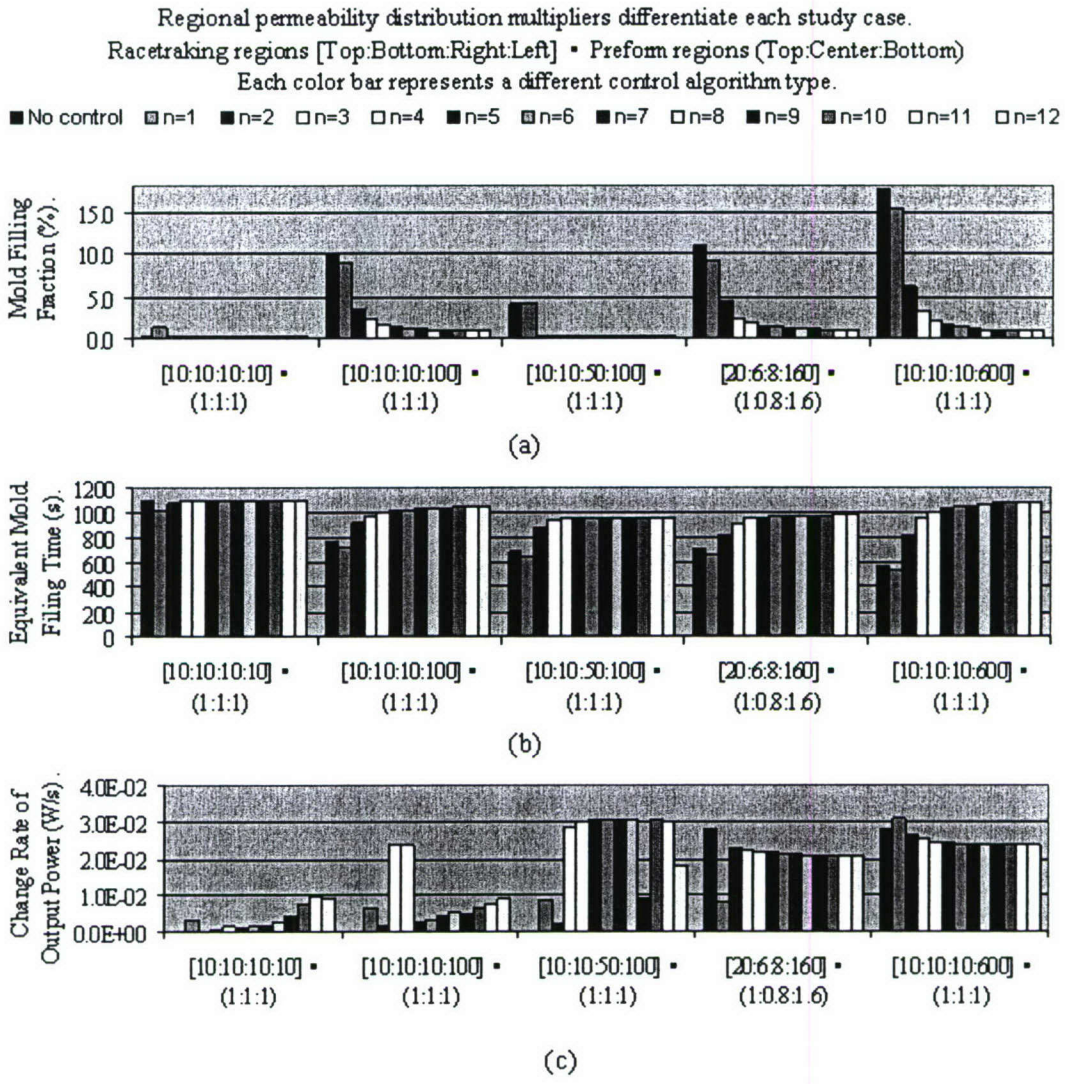


Figure 11. Linearly-corrected inlet pressure driven RTM control process (Equation (7)): (a) comparison of unfilled mold percentage (U), (b) comparison of equivalent filling time (t), and (c) comparison of maximum change rate of pump output power ($\dot{\Pi}$).

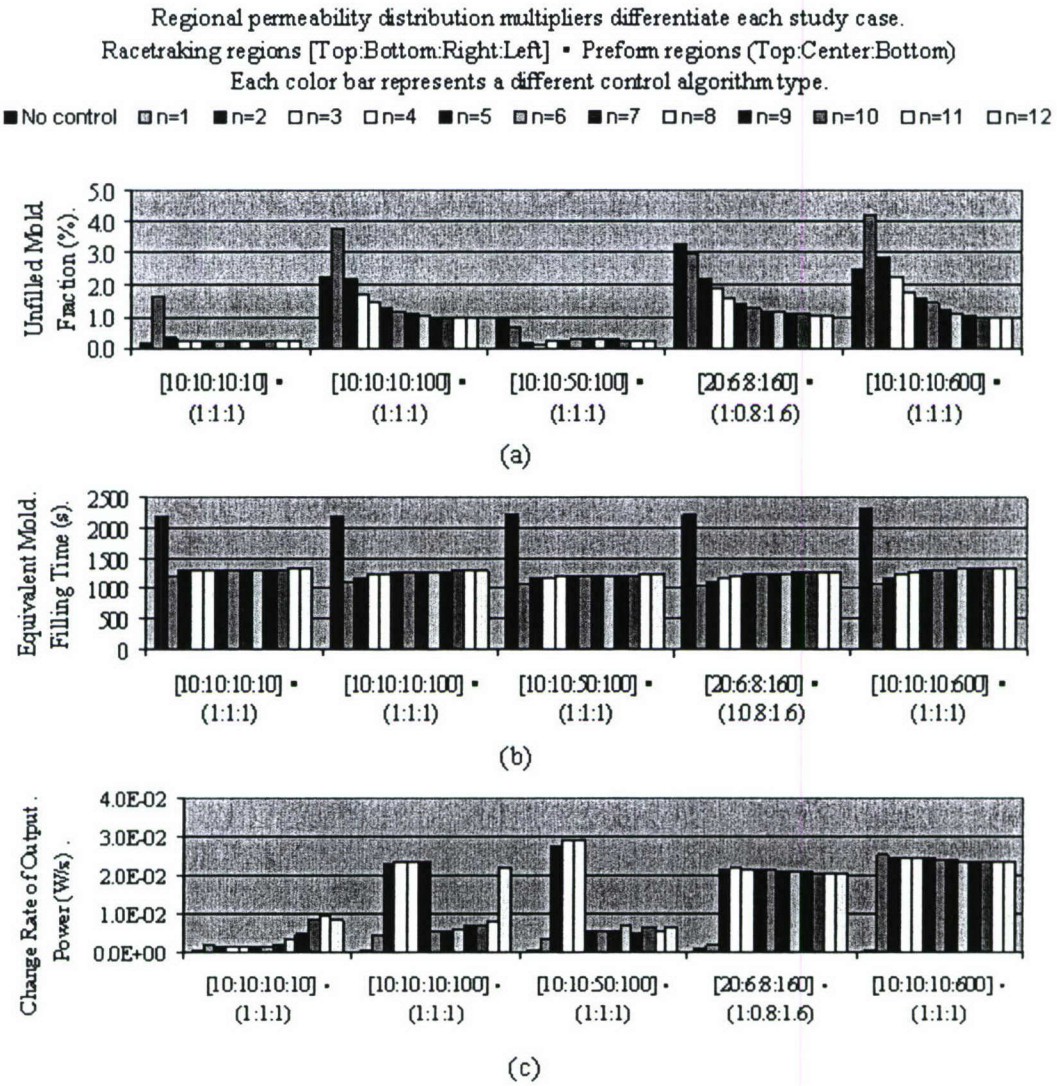


Figure 12. Combined flow rate and linearly-corrected inlet pressure controlled RTM process (Equation (4)) or Equation (7)): (a) comparison of unfilled mold percentage (U), (b) comparison of equivalent filling time (t), and (c) comparison of maximum change rate of pump output power ($\dot{\Pi}$).

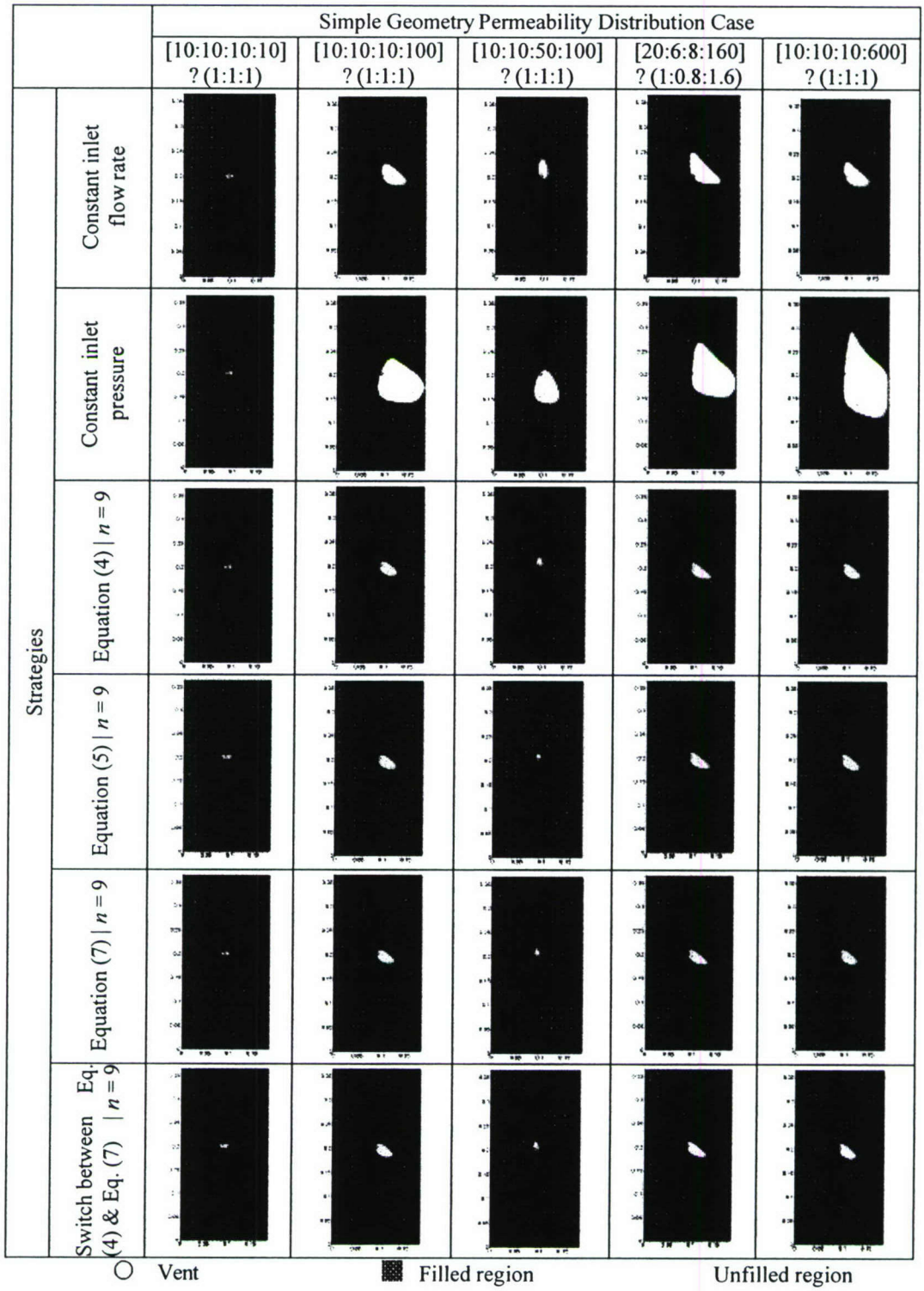


Figure 13. Unfilled area at the instant the first spinal flow front reaches the vent perimeter.

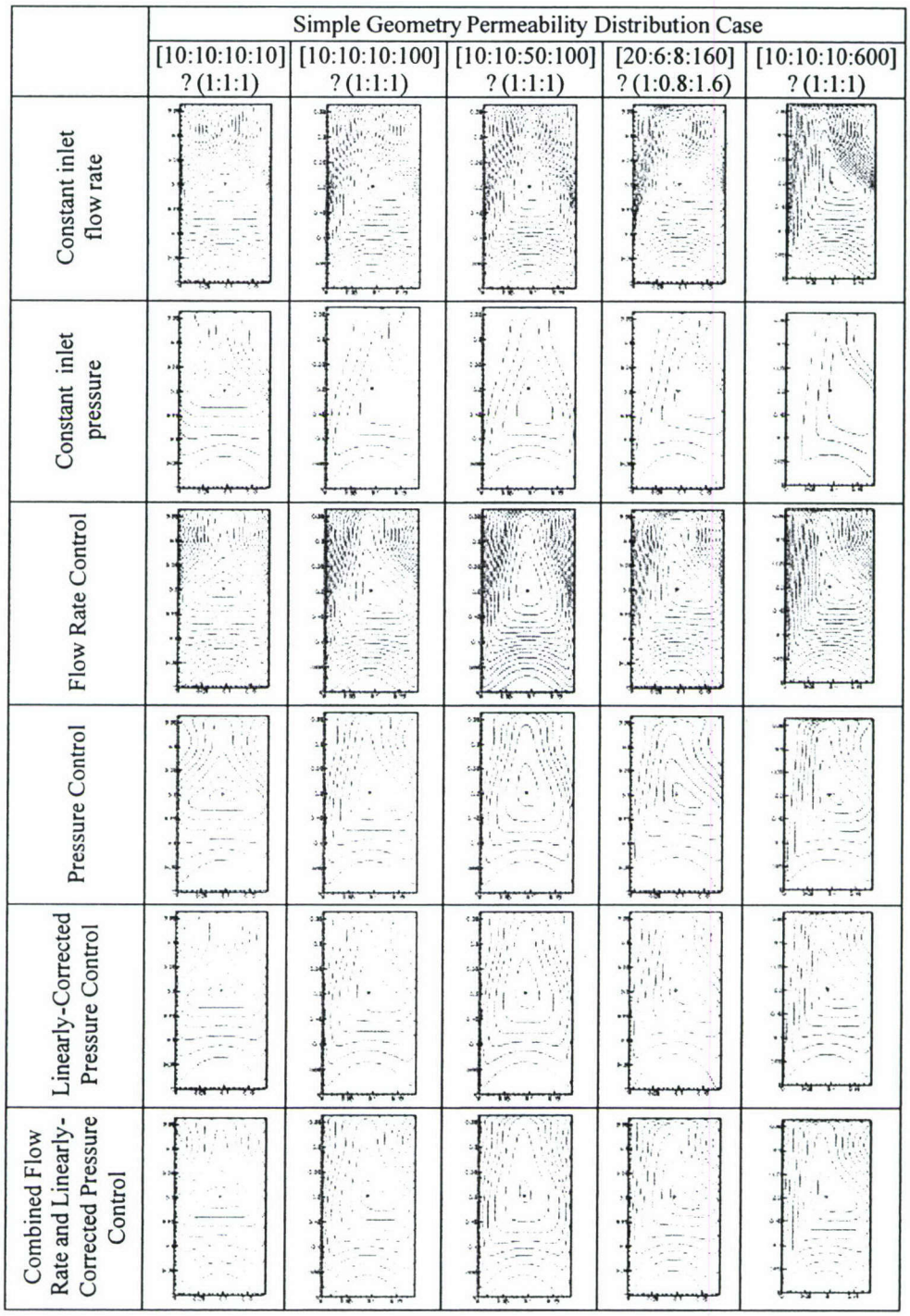


Figure 14. Comparison of flow front contours (*? t*=150 s).

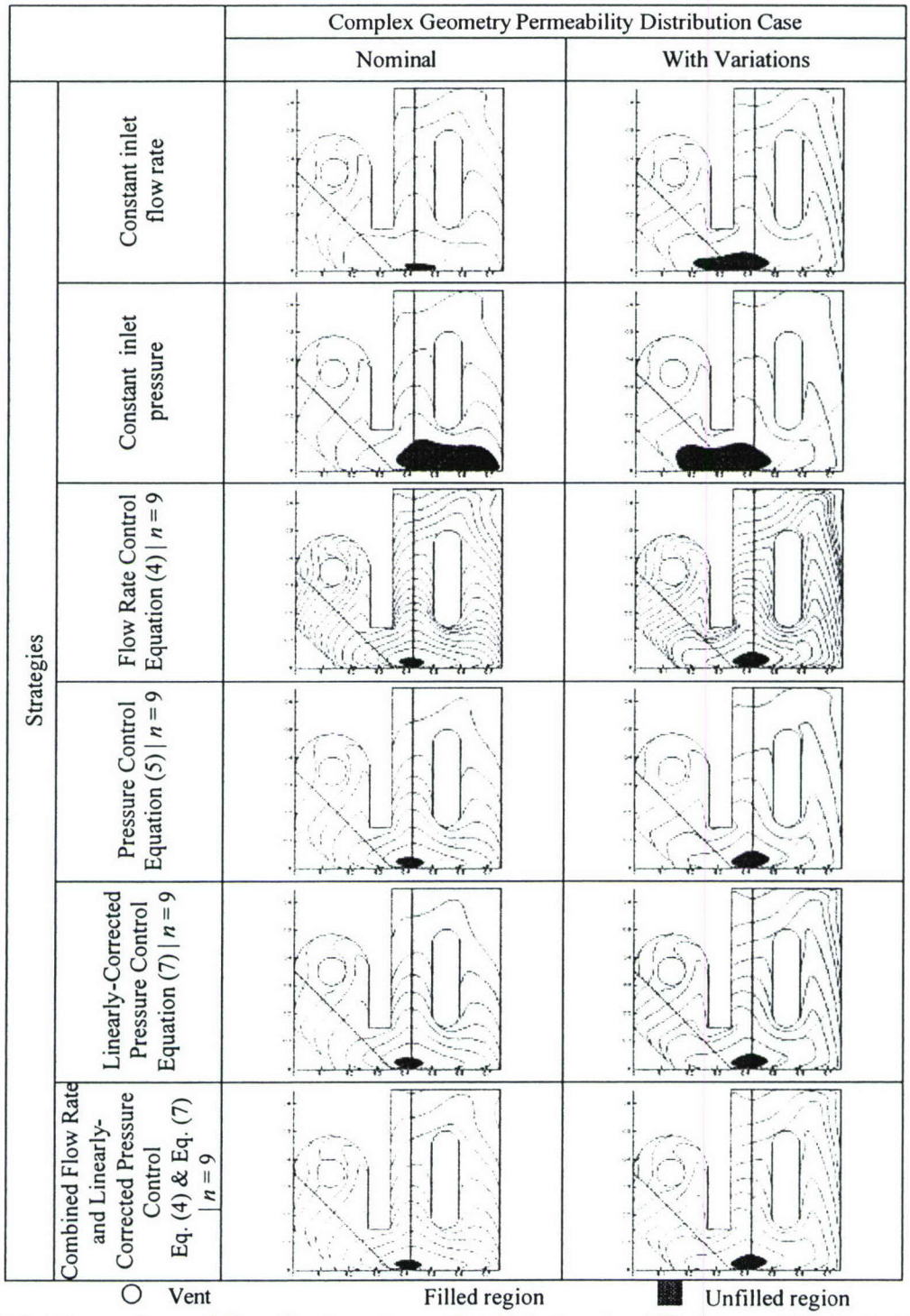


Figure 15. Comparison of flow front contours (? t=100 s) and unfilled area at the instant the first spinal flow front reaches the vent perimeter.

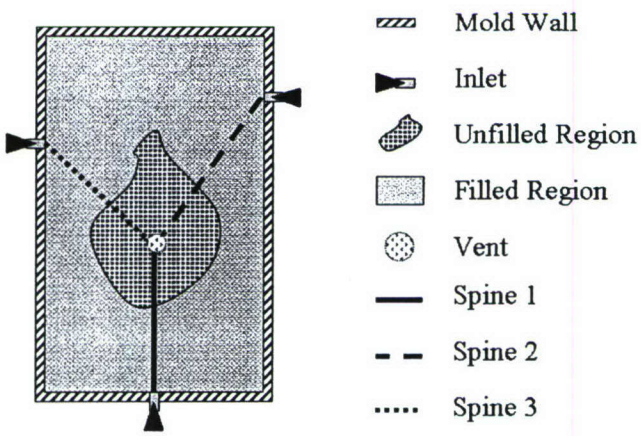


Figure 16. Spinal sensor setup arrangement.

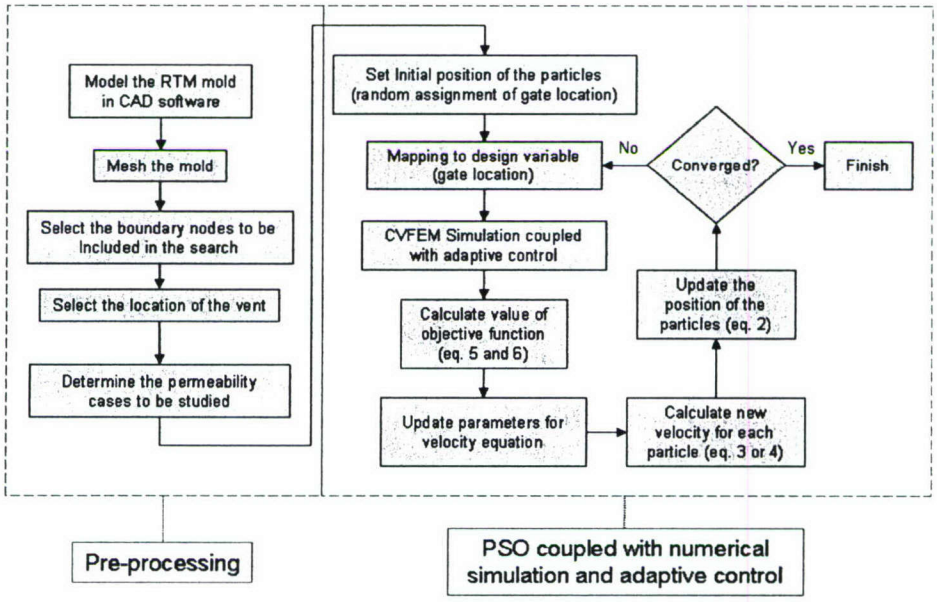


Figure 17. Optimization procedure.

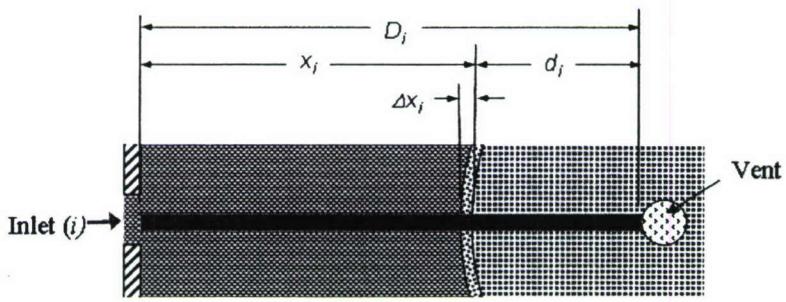


Figure 18. Parameters used in the control method for one line sensor

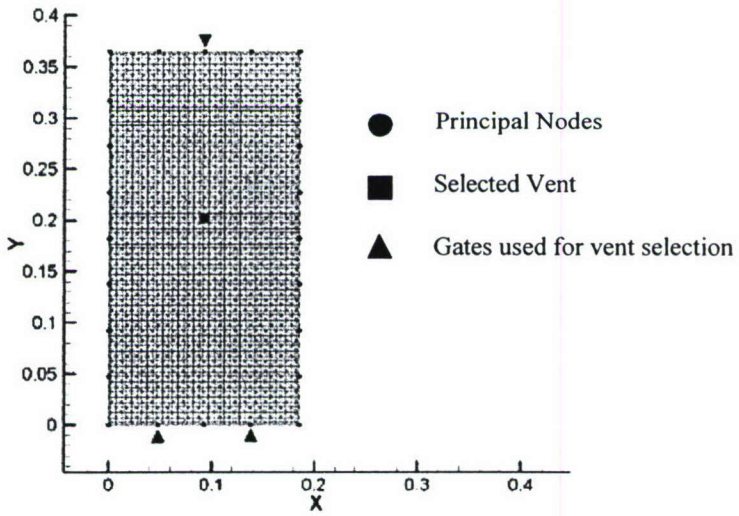
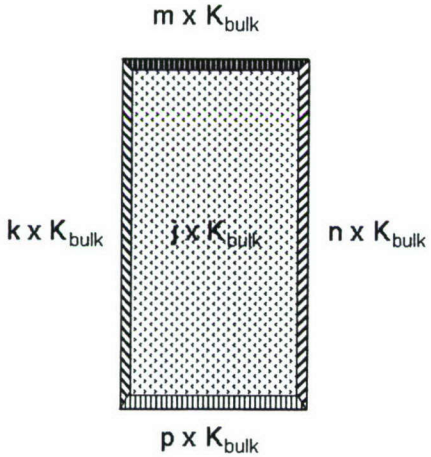


Figure 19. Mesh, principal nodes and vent selection of the case study.

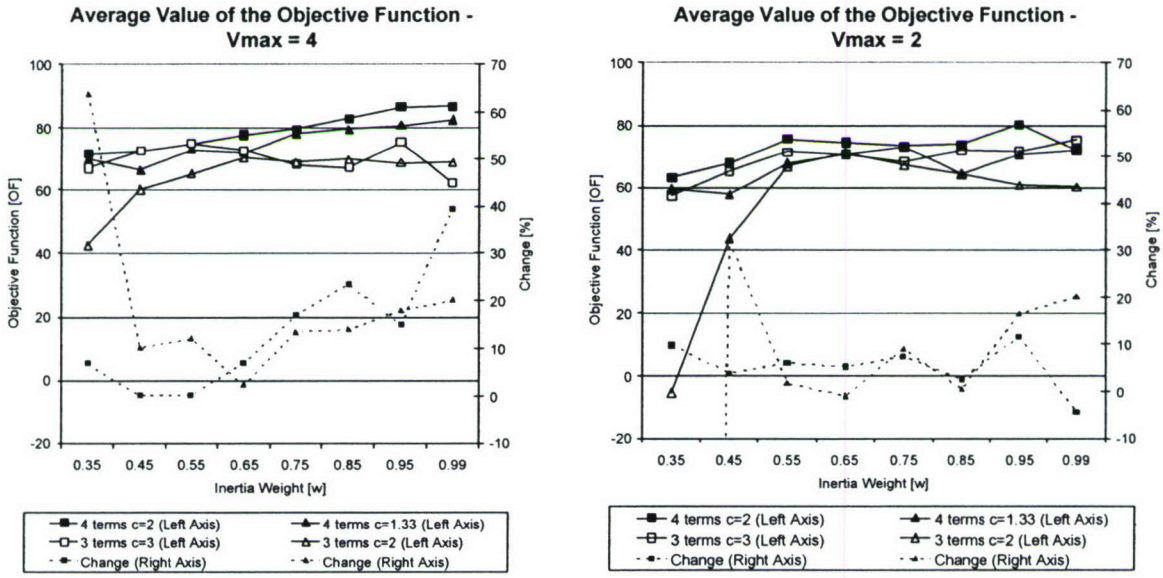
1)



Case	j	k	m	n	p
1	1	10	10	10	10
2	1	30	10	10	10
3	1	10	30	10	10
4	1	10	10	30	10
5	1	10	10	10	30

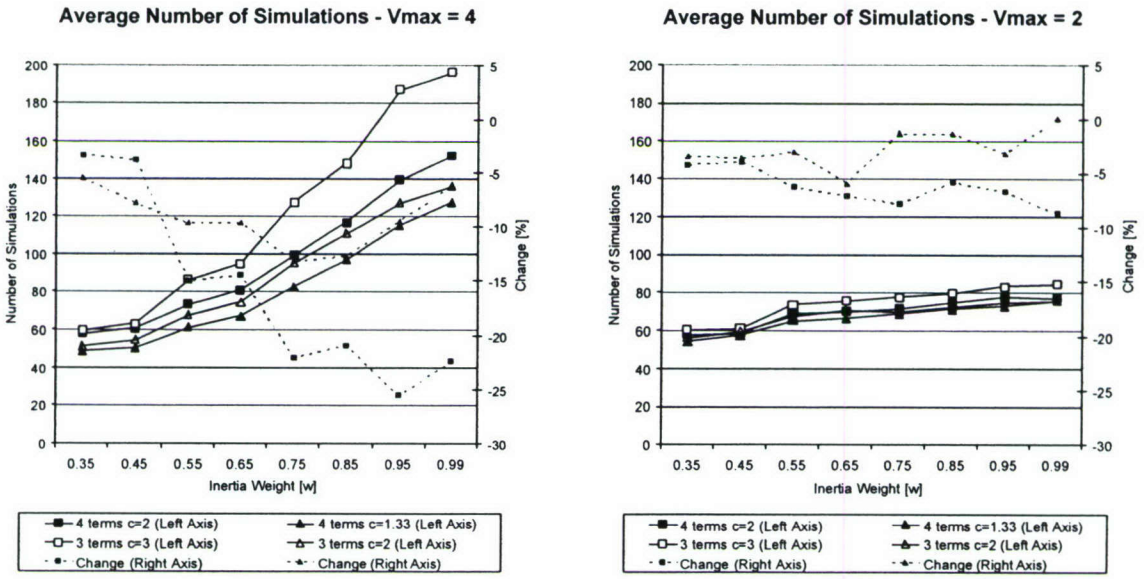
Figure 20. Permeability cases for the case study.

i)



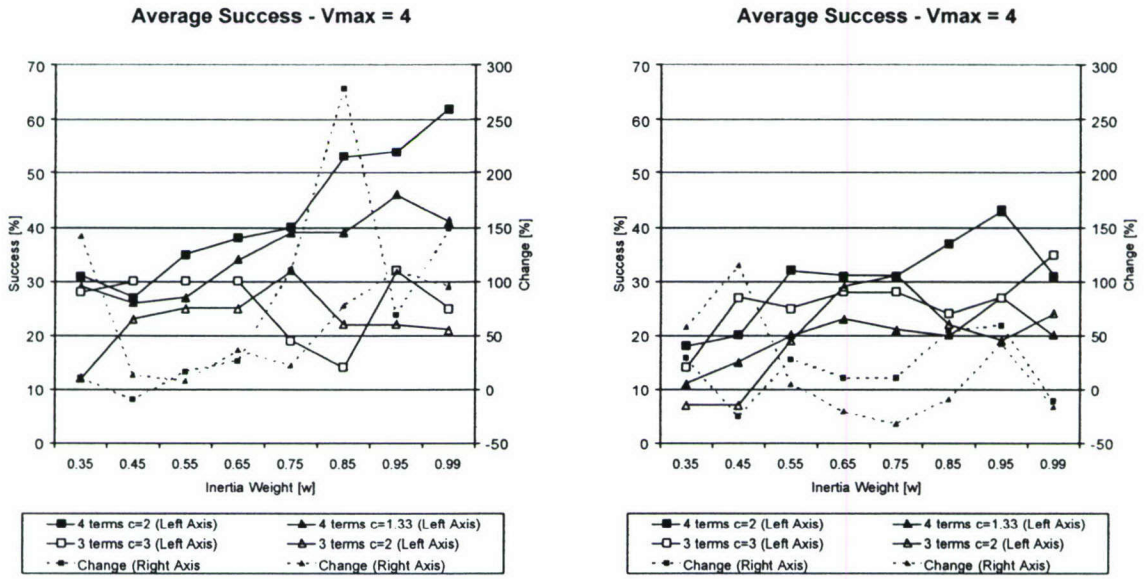
(a) (b)

Figure 21. Average value of the objective function for (a) $v_{\max} = 4$ and (b) $v_{\max} = 2$.



(a) (b)

Figure 22. Average number of simulations for (a) $v_{\max} = 4$ and (b) $v_{\max} = 2$.



(a) (b)

Figure 23. Average number of simulations for (a) $v_{\max} = 4$ and (b) $v_{\max} = 2$.

LIST OF TABLES

Table 1. Equivalent permeability values for different racetracking channel widths.

Permeability Multiplier	Equivalent Permeability (K_β)	Numerical Model Racetracking Channel Width (β)	Effective Physical Racetracking Channel Width (B)
10	$2.0 \times 10^{-10} m^2$	2.5mm	0.185mm
30	$6.0 \times 10^{-10} m^2$	2.5mm	0.262mm
50	$1.0 \times 10^{-9} m^2$	2.5mm	0.311mm
100	$2.0 \times 10^{-9} m^2$	2.5mm	0.391mm
160	$3.2 \times 10^{-9} m^2$	2.5mm	0.458mm
600	$1.2 \times 10^{-8} m^2$	2.5mm	0.711mm

Table 2. Permeability distributions used in the simple geometry numerical case study.

Case #	Permeability Distribution According to Notation	Permeability Multiplier by Region Baseline Permeability = $2.0 \times 10^{-11} m^2$						
		E1	E2	E3	E4	P1	P2	P3
0	[10:10:10:10] ? (1:1:1)	10	10	10	10	1	1	1
1	[10:10:10:100] ? (1:1:1)	10	10	10	100	1	1	1
2	[10:10:50:100] ? (1:1:1)	10	10	50	100	1	1	1
3	[20:6:8:160] ? (1:0.8:1.6)	20	6	8	160	1	0.8	1.6
4	[10:10:10:600] ? (1:1:1)	10	10	10	600	1	1	1

Table 3. The constants used in the objective function $H^{k}_{m(n)}$.

	Unfilled Region Volume	Process Time	Power Change Index
Maximum	$U_{max} = 2\%$	$\tau_{max} = 2500$ s	$\dot{\Pi}_{max} = 3.5$ mW/s
Optimal	$U_{opt} = 0.042\%$	$\tau_{opt} = 1000$ s	$\dot{\Pi}_{opt} = 0.5$ mW/s
Weighting Factor	$F_U = 60\%$	$F_T = 10\%$	$F_{\dot{\Pi}} = 30\%$

Table 4. Permeability distributions used for the complex geometry numerical case study.

Case	Permeability Multiplier by Region Baseline Permeability = $2.0 \times 10^{-11} m^2$																
	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	P1	P2	P3	P4	P5	P6
Nominal Permeability	30	30	30	30	30	30	30	30	30	30	30	1	1	1	1	1	1
With Permeability Variations	30	30	10	30	160	10	30	30	100	160	50	1	1	1	1	1.5	0.9

Table 5. Overall results of the inlet flow rate control algorithm optimization.

Exp. (n)	Control Algorithm Optimization Function $H^{k}_{m(n)}$ Results by Permeability Distribution Case (k), see Equation (8).					Average ($\Xi_{m(n)}$)
	[10:10:10:10] ? (1:1:1)	[10:10:10:100] ? (1:1:1)	[10:10:50:100] ? (1:1:1)	[20:6:8:160] ? (1:0.8:1.6)	[10:10:10:600] ? (1:1:1)	
1	-3.58	-23.48	72.94	-3.52	-6.64	7.15
2	72.18	25.68	83.96	26.53	23.11	46.29
3	80.39	40.55	84.70	35.78	36.56	55.60
4	82.91	47.64	82.89	41.92	41.28	59.33
5	83.91	50.85	82.16	46.24	47.12	62.05
6	84.60	53.04	81.59	49.02	51.47	63.94
7	84.67	54.08	81.35	51.21	53.74	65.01
8	82.00	54.66	81.43	52.51	55.08	65.14
9	81.85	54.83	81.39	53.25	56.20	65.51
10	80.05	54.42	81.06	53.74	56.58	65.17
11	79.31	53.70	81.28	54.40	56.83	65.10
12	78.63	52.27	81.14	54.45	56.60	64.62

Table 6. Overall results of the inlet pressure control algorithm optimization.

Exp. (n)	Control Strategy Optimization Function $H^{k}_{m(n)}$ Results by Permeability Distribution Case (k), see Equation (8).					Average ($\Xi_{m(n)}$)
	[10:10:10:10] ? (1:1:1)	[10:10:10:100] ? (1:1:1)	[10:10:50:100] ? (1:1:1)	[20:6:8:160] ? (1:0.8:1.6)	[10:10:10:600] ? (1:1:1)	
1	41.27	-111.09	25.72	-106.42	-306.88	-91.48
2	93.14	-0.74	95.88	-75.85	-125.92	-2.70
3	97.99	30.51	81.52	-39.55	-62.34	21.62
4	98.77	44.30	17.69	-18.70	-32.18	21.97
5	99.23	49.61	80.47	-9.10	-20.12	40.02
6	97.47	54.04	12.90	-2.21	-10.04	30.43
7	94.76	56.98	11.18	3.79	-5.37	32.27
8	92.30	56.44	9.40	5.60	-2.08	32.33
9	90.09	55.85	61.77	6.98	0.55	43.05
10	88.95	53.97	8.86	7.80	1.65	32.25
11	81.67	54.18	9.03	9.10	2.65	31.33
12	81.52	51.04	9.02	9.97	3.21	30.95

Table 7. Overall results of the linearly-corrected inlet pressure control algorithm optimization.

Exp. (n)	Control Strategy Optimization Function $H^{k}_{m(n)}$ Results by Permeability Distribution Case (k), see Equation (8).					Average ($\Xi_{m(n)}$)
	[10:10:10:10] ? (1:1:1)	[10:10:10:100] ? (1:1:1)	[10:10:50:100] ? (1:1:1)	[20:6:8:160] ? (1:0.8:1.6)	[10:10:10:600] ? (1:1:1)	
1	53.65	-184.96	-43.36	-191.00	-452.15	-163.56
2	98.34	-1.96	90.07	-94.46	-159.63	-13.53
3	99.20	-31.23	15.35	-34.56	-67.97	-3.84
4	96.54	-13.18	9.24	-15.63	-31.32	9.13
5	97.89	59.98	7.79	-3.93	-14.45	29.46
6	96.98	64.16	7.63	2.27	-8.18	32.57
7	95.78	56.55	7.55	7.17	-3.23	32.76
8	93.06	60.79	6.60	9.26	2.22	34.39
9	87.85	63.94	75.82	10.30	3.85	48.35
10	77.85	59.46	6.88	11.62	4.74	32.11
11	71.94	56.06	7.91	12.12	5.21	30.65
12	73.96	53.29	47.66	12.40	5.99	38.66

Table 8. Overall results of the combination of flow rate and linearly-corrected pressure control algorithm optimization.

Exp. (n)	Control Algorithm Optimization Function $H^{k}_{m(n)}$ Results by Permeability Distribution Case (k) , see Equation (8).					Average ($\Xi_{m(n)}$)
	[10:10:10:10] ? (1:1:1)	[10:10:10:100] ? (1:1:1)	[10:10:50:100] ? (1:1:1)	[20:6:8:160] ? (1:0.8:1.6)	[10:10:10:600] ? (1:1:1)	
1	51.68	-22.37	77.62	12.14	-99.72	3.87
2	91.25	-33.06	15.94	-26.41	-58.10	-2.08
3	93.56	-19.47	11.53	-17.76	-40.19	5.53
4	93.74	-11.85	8.68	-9.83	-24.98	11.15
5	93.93	-5.39	82.57	-3.63	-19.78	29.54
6	94.14	55.19	81.21	1.70	-13.50	43.75
7	93.13	54.98	81.68	5.14	-7.20	45.55
8	88.15	55.05	77.61	6.07	-2.46	44.88
9	84.24	54.32	83.84	6.71	0.08	45.84
10	72.51	54.75	79.35	8.14	0.62	43.07
11	69.40	52.01	81.92	8.62	1.54	42.70
12	72.31	8.03	79.39	9.25	2.38	34.27

Table 9. Comparison of the mold filling times (in seconds) of different control algorithms.

Control Algorithm	Results by Permeability Distribution Case				
	[10:10:10:10] ? (1:1:1)	[10:10:10:100] ? (1:1:1)	[10:10:50:100] ? (1:1:1)	[20:6:8:160] ? (1:0.8:1.6)	[10:10:10:600] ? (1:1:1)
Constant Flow Rate	2179.1	2154.5	2197.7	2148.6	2254.0
Constant Pressure	1094.3	691.0	651.9	630.6	471.4
Flow Rate Control	2170.9	2205.7	2220.2	2133.4	2315.5
Pressure Control	1088.0	1010.9	948.2	941.1	1036.1
Linearly-Corrected Pressure Control	1090.4	1025.8	955.0	969.7	1057.2
Combined Flow Rate and Linearly-Corrected Pressure Control	1298.4	1261.7	1214.7	1237.1	1306.7

Table 10. Comparison of the dry spot sizes of different control algorithms.

Control Algorithm	Results by Permeability Distribution Case				
	[10:10:10:10] ? (1:1:1)	[10:10:10:100] ? (1:1:1)	[10:10:50:100] ? (1:1:1)	[20:6:8:160] ? (1:0.8:1.6)	[10:10:10:600] ? (1:1:1)
Constant Flow Rate	0.18%	2.25%	0.91%	3.29%	2.46%
Constant Pressure	0.18%	9.85%	4.00%	11.14%	17.60%
Flow Rate Control	0.20%	1.09%	0.27%	1.24%	1.05%
Pressure Control	0.23%	1.11%	0.20%	1.25%	1.14%
Linearly-Corrected Pressure Control	0.23%	1.01%	0.23%	1.13%	1.01%
Combined Flow Rate and Linearly-Corrected Pressure Control	0.22%	0.99%	0.25%	1.12%	1.02%

Table 11. Comparison of the maximum change rate of pump output power (W/s) required for different control algorithms (n=9).

Control Algorithm	Results by Permeability Distribution Case				
	[10:10:10:10] ? (1:1:1)	[10:10:10:100] ? (1:1:1)	[10:10:50:100] ? (1:1:1)	[20:6:8:160] ? (1:0.8:1.6)	[10:10:10:600] ? (1:1:1)
Constant Flow Rate	5.13E-04	5.13E-04	5.13E-04	6.57E-04	5.13E-04
Constant Pressure	1.72E-07	0.00E+00	1.06E-07	2.76E-02	2.77E-02
Flow Rate Control	1.63E-03	1.36E-03	9.10E-04	7.10E-04	7.47E-04
Pressure Control	3.63E-03	6.30E-03	1.36E-02	2.07E-02	2.34E-02
Linearly-Corrected Pressure Control	4.36E-03	4.59E-03	8.79E-03	2.07E-02	2.35E-02
Combined Flow Rate and Linearly-Corrected Pressure Control	4.68E-03	6.76E-03	4.87E-03	2.07E-02	2.35E-02

Table 12. Complex geometry case study results for different control algorithms (n=9).

Control Algorithm	Permeability Distribution Case			
	Nominal		With Variations	
	Dry Spot Size (%)	Filling Time (sec)	Dry Spot Size (%)	Filling Time (sec)
Constant Flow Rate	0.65%	1322	3.75%	897
Constant Pressure	9.30%	420	8.01%	307
Flow Rate Control	0.61%	1659	1.45%	1509
Pressure Control	0.82%	748	1.70%	512
Linearly-Corrected Pressure Control	0.89%	755	1.76%	527
Combined Flow Rate and Linearly-Corrected Pressure Control	0.72%	909	1.45%	822

Table 13. Constant values for (a) optimization algorithm and (b) numerical simulation.

(a)

2) Maximum Allowable Void (*MAV*)	1%
3) Percentage of Optimal Void (*POV*)	0%
4) Maximum Allowable Time (*MAT*)	100 s
5) Optimal Time (*OT*)	60 s
6) Void Weight (*VW*)	80%
7) Time Weight (*TW*)	20%

(b)

8) Bulk Permeability ($\underline{\underline{K}}$)	2.0×10^{-9} m^2
9) Porosity (ε)	55%
10) Viscosity (μ)	800 Pa·s

Table 14. Comparison between SES [50], simple GA [51], and two and three term PSO.

	Average Value of Objective Function	**Average Number of Simulations**	**Percentage of Success**
SES	96.92	2024	100%
GA	62.09	89.35	15%
Four Term PSO ($w = 0.99,\ c = 2$)	86.81	152.16	62%
Four Term PSO ($w = 0.65,\ c = 2$)	77.71	80.93	38%
Three Term PSO ($w = 0.99, c = 3$)	62.21	196.26	25%
Three Term PSO ($w = 0.65,\ c = 3$)	72.67	94.61	30%